CIRCUIT SIMULATION TECHNIQUES VIA GENERALIZED KRYLOV
SUBSPACES METHODS

by

Emad Gad

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Ottawa-Carleton Institute for Electrical Engineering,
Department of Electronics,
Carleton University

Copyright © 2003 by Emad Gad

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-88718-9

Our file Notre référence

ISBN: 0-612-88718-9

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

The undersigned recommend to the Faculty of Graduate
Studies and Research acceptance of the thesis,

*“Circuit Simulation Techniques via Generalized Krylov
Subspaces Methods”*

submitted by Emad Gad,

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Chairman, Department of Electronics,
Professor Michel Nakhla

Thesis Supervisor,
Professor Michel Nakhla

External Examiner

Professor Ajoy Opal,

Department of Electrical and Computer Engineering
University of Waterloo

Carleton University
May 7, 2003

Abstract

The increasing trends in industry towards integrating various systems on the same chip is putting enormous demands on CAD tools. Circuit simulation is basically the task of representing general circuit structures in the mathematical domain, and then applying numerical analysis techniques to simulate circuit performance. However, the growing complexity in modern circuit designs and device models is rendering many of the conventional analysis techniques impractical. This thesis tackles the problem of increasing complexity of simulation through using reduction approaches. In general, reduction approaches seek to simulate the huge problem in a much smaller space and then bring the solution to the space of the original problem.

The first problem addressed in the thesis is the problem of high-speed analysis of interconnect circuits. Interconnect circuits represent the dominant factor in assessing the performance of high-speed circuit designs. The difficulty in the analysis of interconnect circuits at high-speed stems from the fact that these circuits are naturally described in the frequency-domain, while general circuit simulation is a purely time-domain process. This thesis approaches this problem through using Matrix-Padé approximant to describe the interconnect structure with a macromodel that is amenable to direct time-domain simulation. To increase the efficiency of the time-domain simulation, model-order reduction based on Congruence transform is employed to reduce the size of the total circuit. In addition, a rigorous proof of the passivity of the reduced-order model is presented to guarantee the stability of the numerical simulation process.

The second proposed solution to manage the increasing complexity in modern designs is geared towards the problem of finding the steady-state solution for large nonlinear circuits. Finding the steady-state solution is an essential requirement for the design automation process of analog circuits. However, for large circuits excited by multi-tone inputs, this problem becomes intractably huge. The proposed approach is based on Krylov-subspace projection

methods to project the original large problem onto a much smaller subspace. The problem is then solved entirely on the reduced space and the solution obtained is then mapped back to the original space of the problem. The reduction in the size of the problem results in a very significant saving in computational complexity.

The thesis also presents a new algorithm for abstraction of large linear time-varying systems into reduced-order macromodels. Linear time-varying systems appear in many contexts during circuit simulation. For example, noise behavior in analog nonlinear circuits is always described by linear time-varying systems. The approach of the proposed algorithm is motivated by the fact that the transfer functions of linear time-varying systems is characterized by short transient response in the right-half plane of the Laplace-domain. Obtaining reduced-order approximation for the transfer function can be achieved easily using simple numerical time-domain integration for a very short period of simulation time. The proposed algorithm therefore avoids completely the need to solve large dense linear systems of equations, which results in saving of several orders-of-magnitude in computational complexity.

Based on the idea of model-reduction of linear time-varying systems, the thesis also tackles the problem of simulating nonuniform transmission lines through Integrated Congruence Transform (ICT). The algorithm proposed describes an orthogonalization procedure for a set of elements related through a differential operator without having to compute these elements explicitly.

Dedication

To my parents and my wife.

Acknowledgements

It has never been an easy task trying to dedicate a single page to mention and thank all those wonderful people I came to know during my journey that stretched over five years of exciting work at Carleton. What makes things more difficult is trying to harness words to convey deep feelings of gratitude.

At the forefront of those whose support and mentorship presented the major driving force to bring this work into completion, comes professor Michel Nakhla. On both the technical and personal levels, professor Nakhla's support has been pivotal in shaping my goals and focus for the rest of my life. I owe him the passion for research that he sparked into my heart.

I feel also so much indebted for my colleagues. The enthusiasm and collaboration spirit that permeated our group during those five years have engraved nice memories for a long time to come. In particular, Dr. Roni Khazaka, Anestis Dounavis, Dr. Pavan Gunupudi and Richard Griffith have provided me with remarkable time through mutual projects and nice conversations. Other colleagues such as Dharmendra Saraswat, Christina Oregon, and Ihsan Irdin have also been instrumental in creating a very friendly environment.

I would also like to express a deep appreciation to my uncle Maher Awad. Through an indelible smile and unconditional love, he put my feet on this road.

Also my deep appreciation for my parents and especially my mother who has put all her faith in me.

My wife is also one more central person whose love, support, encouragement and understanding have nurtured many parts and aspects of the work done in the thesis. She has been truly capable of bringing together an exceptional personal dedication in helping me, with a phenomenal technical support through superb "LaTeX and programming skills" to write this thesis in a record time.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Contributions	3
1.3	Background and Overview	4
1.3.1	Transient Analysis of Linear Distributed Elements	5
1.3.2	Passivity	6
1.3.3	Matrix-Pad� Modelling of Uniform Transmission lines	6
1.3.4	Model-Order Reduction	7
1.3.5	The Arnoldi Process	8
1.3.6	Steady-State Analysis of Nonlinear Circuits	9
1.3.7	Model Reduction of Linear Periodically Time Varying Systems	13
1.3.8	Simulation of Nonuniform Transmission Lines	14
1.4	Organization of the Thesis	15
2	Generalized Formulation of Circuit Equations	16
2.1	MNA Formulation of Linear Circuits	16
2.2	Properties of the MNA Matrices	17
2.3	Nonlinear Elements	20
2.4	MNA Formulation for Transmission Lines	22

2.4.1	Matrix Exponential Transmission Line Stencil	24
2.4.2	General MTL Formulation	26
2.4.3	An Example of MTL Formulation	26
2.5	Mixed Frequency-Time Problem	28
3	Background on Steady-State Analysis Techniques	29
3.1	Time-Domain Approaches	30
3.1.1	Conventional Transient Analysis	30
3.1.2	Shooting Method	32
3.2	Harmonic Balance Approach	35
3.2.1	HB for Linear Circuits	35
3.2.2	HB for Nonlinear Circuits	36
3.2.3	An Example of Harmonic Balance Equations	37
3.2.4	General Formulation of the Harmonic Balance Equations	38
3.2.5	Solving the Harmonic Balance Equations	39
3.2.6	Iterative Methods	47
3.3	Quasi-Periodic Steady-State Response	51
4	Simulation Techniques of High-Speed Interconnects	53
4.1	Passivity Perseveration	54
4.2	Time-Domain Macromodels	55
4.2.1	RLC Lumped Networks	56
4.2.2	Chebyshev Based Macromodels	57
4.2.3	Matrix-Pad� Macromodel	58
4.3	Model-Order Reduction	64
4.3.1	Explicit Moment Matching	64
4.3.2	Implicit Moment Matching	67

4.3.3	MOR Based on Integrated Congruence Transform	69
4.4	Discussion	73
5	Passive Model Reduction of Distributed Interconnects	77
5.1	Development of MNA Stamp	78
5.2	Model Reduction and Passivity Preservation	83
5.2.1	Model Reduction	83
5.2.2	Passivity Preservation	84
5.3	Computational Results	91
5.3.1	Example 1	92
5.3.2	Example 2	93
5.4	Conclusion	96
6	Finding Steady-State Solution via Circuit Reduction	101
6.1	Introduction	101
6.2	Solving Nonlinear equations	103
6.2.1	Newton-Raphson (NR) Method	104
6.2.2	Continuation Methods	104
6.3	Overview of the Circuit reduction Algorithm	106
6.3.1	Circuit Reduction	107
6.3.2	Solution of the Reduced Equations	109
6.4	Calculation of the Subspace Q_h	111
6.5	Computational Complexity	115
6.5.1	Moments Computation	115
6.5.2	Construction of the Reduced Jacobian	118
6.6	Proof of Preservation of Derivatives	119
6.7	Numerical Examples	125

6.7.1	Example 1	125
6.7.2	Example 2	129
6.7.3	Example 3	130
7	MOR of Linear Periodically Time Varying Systems	134
7.1	Introduction	134
7.2	Time-Varying System Function	137
7.3	Review of MOR for LPTV Systems	140
7.3.1	Frequency-Domain Matrix Form	142
7.3.2	Time-Domain Matrix Form	143
7.3.3	Model-Reduction Using Block-Krylov Methods	144
7.3.4	Difficulties in Algebraic-Based Model Reduction	145
7.4	Proposed Algorithm	147
7.4.1	Elements from Hilbert Space	147
7.4.2	Construction of LTV Reduced-Order Macromodel	149
7.4.3	Moment Matching Theorem	152
7.5	Orthonormal Basis Construction	154
7.5.1	Basis Construction via Implicit Moment Computation	155
7.5.2	Transient Compression	156
7.5.3	Eliminating Numerical Noise	159
7.5.4	Summary	160
7.6	Numerical Results	165
7.6.1	Diode Circuit	165
7.6.2	Mixer Circuit	166
7.6.3	Mixer Circuit with strong nonlinearity	167
7.7	Conclusion	167

8	Simulation of Nonuniform TL's via ICT	175
8.1	Outline of the Proposed Algorithm	176
8.2	Implementation Details	180
8.3	Proof of (8.15)	184
8.4	Numerical Results	187
8.4.1	Example 1	187
8.4.2	Example 2	187
8.4.3	Example 3	190
9	Summary and Future Work	193
9.1	Summary	193
9.2	Future Work	194
	Bibliography	197

List of Tables

5.1	CPU comparison of time-domain response for example 1.	92
5.2	CPU comparison of time response for example 2.	93
6.1	Formulas for the derivatives of some simple functions	114
7.1	CPU Computational Comparison	168
8.1	CPU time comparison	192

List of Figures

2.1	Component Stamps in MNA formulation.	18
2.2	Illustration of the formation of the MNA matrices for a two-port RLC circuit example.	19
2.3	A rectifier circuit with a nonlinear inductor.	21
2.4	High-Speed interconnects.	23
2.5	Schematic illustration for an MTL transmission line.	24
2.6	A circuit example containing a 2-Conductors transmission line.	27
3.1	An example circuit for HB formulation.	37
3.2	Sparsity patterns of $\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$ for different degrees of nonlinearity (Single-Variable case).	44
3.3	Sparsity pattern of $\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$ for different degrees of nonlinearity (Multi-variables case).	47
3.4	Pseudo-code description for the Arnoldi algorithm.	49
4.1	Interconnect cross-section and wavelength of the propagating signal.	54
4.2	Passivity preservation of a macromodel connected to a passive termination . . .	55
4.3	Electrical interconnect models	56
4.4	Multiconductor transmission line represented by subsections of Padé model. . .	61
4.5	Pseudo-code description of Block Arnoldi process.	69

4.6	Pseudo-code description of MGS process.	72
4.7	MOR using PRIMA algorithm with <i>explicit</i> moment matching.	74
4.8	MOR using PRIMA algorithm with <i>implicit</i> moment matching.	75
5.1	Realization of the Y -parameters in (5.1) of a 1-conductor transmission line. . .	79
5.2	Realization of the Y -parameters in (5.2) of a 1-conductor transmission line. . .	79
5.3	A Resistor's stamp.	82
5.4	A linear subnetwork circuit with nonlinear termination.	93
5.5	Magnitude of Y_{11} of linear subnetwork for example 5.3.1	94
5.6	Magnitude of Y_{12} of linear subnetwork for example 5.3.1	94
5.7	Magnitude of Y_{22} of linear subnetwork for example 5.3.1	95
5.8	Time domain response at output node V_{p2} for example 5.3.1	95
5.9	Time-domain response at output node V_{out} for example 5.3.1	96
5.10	R and L vs. frequency for example 5.3.2	96
5.11	Magnitude of Y_{11} of linear subnetwork for example 5.3.2	97
5.12	Magnitude of Y_{12} of linear subnetwork for example 5.3.2	98
5.13	Magnitude of Y_{22} of linear subnetwork for example 5.3.2	98
5.14	Time response at output port V_{p2} for example 5.3.2	99
5.15	Time domain response at output node V_{out} for example 5.3.2	100
6.1	Schematic representation for the proposed algorithm.	107
6.2	Schematic representation for computing the moments in (6.54).	117
6.3	Sparsity pattern of $\mathcal{D}\bar{f}$	119
6.4	Example 1: Solution at the first harmonic.	126
6.5	Example 1: Solution at the third harmonic.	127
6.6	Example 1: Solution at the fifth harmonic.	127
6.7	Periodic time-domain response of the bipolar circuit (Example 1).	128

6.8	Example 2: Comparison of intermodulation tones.	129
6.9	OpAmp circuit of example 3.	131
6.10	Example 3: Amplitude of the second harmonic.	132
6.11	Example 3: Amplitude of the fifth harmonic.	132
6.12	Example 3: Amplitude of the 9 th harmonic.	133
6.13	Example 3: Harmonics at terminus of continuation sweep.	133
7.1	Typical response of a periodically time-varying system of frequency ω_0	139
7.2	Low-Pass filter→mixer→two bandpass filters	139
7.3	$ H_1(s) $ for the circuit of Figure 7.2. ($R_1 = 160\Omega$, $R_2 = 1.6k\Omega$, $C_1 = C_2 =$ $C_3 = 10nF$, $L_2 = L_3 = 25.35nH$, $\omega_0 = 10MHz$)	140
7.4	Frequency response of the bandpass mixer of Figure 7.2.	141
7.5	Steady-state response for the system (7.27), at $q = 0$ (upper graph) and $q = 20$ (lower graph)	146
7.6	Pseudo-code description for generating the orthonormal basis $Q(t)$	161
7.7	A rectifier circuit.	166
7.8	Frequency response of diode circuit	169
7.9	A mixer Circuit	170
7.10	Steady-state response for the mixer circuit.	171
7.11	Transient response for the system in (7.51).	172
7.12	Frequency response of mixer circuit.	173
7.13	Relative percentage error in $ H_1(s) $ obtained from the reduced system.	174
8.1	A comparison for the $ Y_{11} $ parameter between the proposed algorithm the al- gorithm in [1] using explicit moments computation.	188
8.2	Numerical results for a 10 cm nonuniform transmission line. (Frequency- domain Response)	189

8.3	Numerical results for a 10 cm nonuniform transmission line. (Time-domain Response)	190
8.4	Numerical results for a 50 cm nonuniform transmission line	191
8.5	Numerical results for a 50 cm nonuniform transmission line	192

Notation

A	A real matrix A
A^T	A real matrix A transposed
A^{-T}	Matrix A inverted and transposed
A^*	Complex conjugate of complex matrix A
F^{T*}	Complex conjugate and transpose of a complex vector F
$\mathbb{R}^N$	The set of real vectors of size N
$\mathbb{R}^{N \times N}$	The set of real matrices of size $N \times N$
$\mathbb{C}^N$	The set of complex vectors of size N
$\mathbb{C}^{N \times N}$	The set of complex matrices of size $N \times N$
$d_{i,j}$	Matrix composed of elements $d_{i,j}$ in i^{th} row and j^{th} column
$I(x, s)$	Laplace-domain current at distance x for a distributed network.
$V(x, s)$	Laplace-domain voltage at distance x for a distributed network.
U	The identity matrix
$\mathcal{L}_p(T)$	The Hilbert space of all t -dependent vectors of size p in the interval $0 \leq t \leq T$.
$\mathcal{L}^{2m}(0, d)$	The Hilbert space of all z -dependent vectors of size $2m$ in the range $0 \leq z \leq d$.
$U_{m \times m}$	An $m \times m$ identity matrix.
I_m	An $m \times m$ identity matrix.

Abbreviations

APFT	Almost Periodic Transform
CAD	Computer Aided Design
CFH	Complex Frequency Hopping
CPU	Central Processing Unit
DAE	Differential-Algebraic Equation
DFT	Direct Fourier transform
EDA	Electronic Design Automation
EMM	Explicit Moment Matching
FFT	Fast Fourier Transform
HB	Harmonic Balance
ICT	Integrated Congruence Transform
IDFT	Inverse Direct Fourier transform
IFFT	Inverse Fast Fourier transform
IMM	Implicit Moment Matching
LMS	Linear Multi-Step
LPTV	Linear Periodically Time-Varying
LTi	Linear Time Invariant
LTV	Linear Time-Varying
MCM	Multi-Chip Modules

MDFT	Multi Dimensional Fourier Transform
MGS	Modified Gram-Schmidt
MMIC	Monolithic Microwave Integrated Circuits
MNA	Modified Nodal Admittance
MOR	Model-Order Reduction
MTL	Multi-conductor Transmission Lines
MVP	Matrix-Vector Product
NR	Newton Raphson method
PCB	Printed Circuit Board
PUL	per unit length
RC	Resistor-Capacitor network
RF	Radio Frequency
RFIC	Radio Frequency Integrated Circuits
RLC	Resistor-inductor-capacitor network
TE	Truncation Error
TEM	Transverse Electromagnetic Mode
TVS	Time-Varying systems
VLSI	Very Large Scale Integrated circuits

Chapter 1

Introduction

1.1 Motivation

The recent rapid development in device fabrication and the increasing propensity towards more complex integrated designs operating at higher frequency have placed an extraordinary demand on *Electronic Design Automation* (EDA) tools. The importance of EDA tools in the design cycle stems from the need to reduce the time-to-market factor of the development cost. This need is further highlighted by the continuous shrinkage of market windows.

The case of *Very Large Scale Integrated* (VLSI) circuits high-speed interconnects is a typical example that signifies the importance of using *Computer Aided Design* (CAD) tools. The recent trends in VLSI industry towards decreasing feature sizes and using mixed analog/digital signal circuits operating at increasing higher frequencies have made signal integrity a challenging task [2]. In these designs, the electronic performance of chip and *Printed Circuit Board* (PCB) interconnects become critical in the overall performance of the circuit design. These interconnects can no longer be considered as simple short circuits but must be treated as distributed *Multi-conductor Transmission Lines* (MTL) [3]. Therefore, phenomena associated with transmission lines such as delays, dispersion, attenuation, ringing and cross talk due to

coupling between lines must be predicted accurately [4]. Furthermore, the proportion of delay time in signal propagation due to interconnect effects is becoming more dominant as the devices are reduced in size and gate delay is further reduced. The objective of employing EDA tools in the design process is therefore the detection of these interconnect effects at early design stages to avoid repeating design cycles unnecessarily.

Another area that has been the scene of increasing complexity in circuit design is the area of analog circuit design. This can be attributed to the current surge in the market of mobile communication equipments which has led to an increasing interest in certain types of circuits such as *Radio Frequency Integrated Circuits* (RFIC) [5].

Many factors have contributed to the increasing complexity of such designs. One of these factors is the drive towards higher integration levels with the objective of integrating more circuit blocks to achieve a totally integrated front-end. Another factor is the increasing complexity of the device models which arises as designers explore more advanced technologies such as CMOS. Increasing the complexity of the device model further increases the number of internal nodes and circuit variables that have to be taken into account during the simulation process, thus increasing the problem size. In addition, the problem of finding steady-state solution of nonlinear circuits is a challenging task.

The objective of this thesis is to address the increasing demands on EDA tools via reducing the computational complexity associated with the analysis and simulation of large circuits. The proposed algorithms approach the problem through model and circuit-reduction techniques.

The first proposed algorithm addresses the problem associated with the analysis of high-speed interconnects through model-reduction techniques based on Krylov-subspace methods [6], [7], [8]. The algorithm has been proved to produce a guaranteed passive reduce-order models. Passivity is a pivotal requirement in all representations of high-speed interconnects in order to ensure stable numerical simulation.

The second proposed algorithm addresses the problem of finding the steady-state solution

through the introduction of the concept of reduction of nonlinear equations [9], [10]. The proposed algorithm enables the huge number of nonlinear equations that result during steady-state analysis to be reduced to a much smaller set of nonlinear equations. The proposed algorithm then proceeds by applying continuation schemes to find the solution of the original large system via computing the solution from the reduced space and mapping it back to the space of the original system. The computational advantage here is that only a small set of equations needs to be solved as opposed to the original huge system. In addition, using continuation methods as a vehicle for solving nonlinear equations greatly improves convergence properties.

The thesis also introduces a new algorithm for abstraction and Model-Order Reduction (MOR) of large systems for the purpose of small-signal analysis. This type of analysis requires dealing with *Linear Periodically Time-Varying* (LPTV) systems. Model-order reduction of LPTV systems in circuit applications has been proposed in [11], [12]. In contrast to these approaches, the proposed algorithm does not require the solution of a large system of equations. Instead, it uses the idea of compressed transient system function to construct reduced-order models by expanding the transfer function in the right-half plane of the Laplace-domain.

1.2 Contributions

In this thesis, generalized projection techniques are developed to address the problem of steady-state analysis of nonlinear circuits and VLSI high-speed interconnects. Specifically, the main contributions are:

1. A new algorithm for reducing nonlinear equations was developed to find the steady-state solution of analog microwave nonlinear circuits [9], [10]. Model-Order Reduction has been traditionally applied to linear circuits reduction. The algorithm presented in this thesis extends that approach for nonlinear circuits. This algorithm is presented in Chapter 6.

2. Development of a guaranteed passive reduced-order model based on the Matrix-Padé time-domain macromodelling technique of high speed interconnects [6], [7], [8]. The proposed algorithm uses orthogonal-projection techniques to reduce large size networks consisting of both lumped and distributed elements into a smaller network with much more reduced-order. Chapter 5 presents the proposed algorithm and a rigorous proof of the passivity of the reduced system.
3. A new algorithm for model-order reduction of LPTV system is provided in Chapter 7. The proposed algorithm uses the idea of compressed transients of the time-varying transfer functions in the right-half plane of the Laplace-domain to avoid having to solve a large system of linear equations [13].
4. Development of an orthogonalization process for a set of elements that are related through a differential operator. The computational advantage of the proposed orthogonalization process is that it does not require these elements to be available explicitly, which greatly improves the accuracy of the resulting orthogonal basis. The proposed orthogonalization process is explained in details in Section 7.5. This process has been employed in obtaining reduced-order approximations of LPTV systems.
5. A novel technique for simulation of nonuniform transmission line networks based on the new orthogonalization process and the concept of Integrated Congruence Transform. This contribution is outlined in Chapter 8.

1.3 Background and Overview

This section presents a quick background on the main topics discussed in the thesis. In addition, it provides a general overview of the basic problems related to the CAD area that is addressed using the proposed algorithms in this work.

1.3.1 Transient Analysis of Linear Distributed Elements

The problem of accurately predicting the effect of physical interconnects on the transient behavior of digital pulses can be cast into that of performing transient analysis of linear distributed circuits. While interconnects are ideally designed to function as simple shorts, whose effects can be neglected, their presence has to be fully taken into account when operating at sufficiently high frequencies. A simple way of modelling the effects of interconnects is to represent the transmission line with a number of sections of linear lumped components of resistors, inductors, and capacitors, (RLC modelling) that can be easily coupled with conventional simulators such as SPICE [3]. However, at higher operating frequencies, lumped-element representation becomes inadequate, and fully distributed transmission lines with per unit length parameter matrices are needed to model interconnects.

Mathematical representation of distributed models is usually obtained through the Telegrapher's equations [3]. Being a set of partial differential equations, the Telegrapher's equations introduce significant challenges as they do not have a direct representation in the time-domain. Thus, simple transient simulation of interconnects effects based on the Telegrapher's equations is not possible in the presence of nonlinear terminations, since nonlinear elements have only a time-domain representation. This problem is known in the literature as the mixed time-frequency problem.

Discretization along the spatial variable in the Telegrapher's equations has been suggested to represent distributed elements in a form amenable to time-domain simulation [14]. Although lumped RLC segmentation can be considered as the simplest form of transmission line discretization, other discretization techniques, e.g. [14], can be more efficient in the sense that the number of extra variables needed is reduced.

1.3.2 Passivity

Another important issue that needs to be addressed in the context of obtaining efficient time-domain macromodels of distributed elements is the issue of passivity [15]. It should be stressed here that although a general transmission line is a passive linear structure by nature, its mathematical representation, based on various representations techniques of the Telegrapher's equations need not be passive [4].

The issue of passivity plays a central role in the numerical stability of time-domain simulation. The reason for that may be stated briefly as follows. It is well known from basic circuit theory that connecting a passive network to another passive network is asymptotically stable, in the sense that the poles of the whole system lies in the left-half plane of the Laplace-domain. However, connecting two non-passive networks is not guaranteed to be stable. For example, having a non-passive representation of some transmission line terminated with non-passive elements can lead to a total unstable network. It is worth mentioning here that this type of instability is an artificial one, since it only appears during the numerical time-domain simulation and not as a result of some instability in the network being simulated.

1.3.3 Matrix-Pad Modelling of Uniform Transmission lines

The matrix-Pad approximation of the Telegrapher's equations has been suggested in [16] to overcome the mixed time-frequency problem associated with the transient simulation of transmission lines. The basic idea in this technique is to replace the state-transition matrix for the uniform transmission line, which happens to be the matrix exponent of the *per unit-length* (PUL) parameters, with a matrix rational function. This enables the Telegrapher's equations to be represented in a form that can be easily plugged in general time-domain simulators such as SPICE [17]. In addition, this technique offers other several advantages. For example, the time-domain representation is obtained in a closed-form without the need to carry out some

preprocessing techniques. Another advantage of the matrix-Padè model is its ability to preserve the passivity of the transmission line's structure.

1.3.4 Model-Order Reduction

The concept of Model-Order Reduction (MOR) aims at replacing a large Multi-Input Multi-Output linear system of m input ports and n output ports, represented with a set of differential equations of size N :

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \quad (1.1)$$

$$\mathbf{y}(t) = \mathbf{C}^T \mathbf{x}(t) \quad (1.2)$$

with another reduced system of size q :

$$\frac{d\mathbf{z}(t)}{dt} = \hat{\mathbf{A}}\mathbf{z}(t) + \hat{\mathbf{b}}u(t) \quad (1.3)$$

$$\tilde{\mathbf{y}}(t) = \hat{\mathbf{C}}^T \mathbf{z}(t) \quad (1.4)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\hat{\mathbf{A}} \in \mathbb{R}^{q \times q}$, $\mathbf{b} \in \mathbb{R}^{N \times m}$, $\hat{\mathbf{b}} \in \mathbb{R}^{q \times m}$, $\mathbf{C} \in \mathbb{R}^{N \times n}$, $\hat{\mathbf{C}} \in \mathbb{R}^{q \times n}$. The reduction becomes useful if $\tilde{\mathbf{y}}(t) = \mathbf{y}(t)$, with $q \ll N$ [18].

MOR approaches can be classified into two main categories, the first category is based on explicit moment matching [19], while the other category is based on Krylov subspace methods [20], [21]. The second category has the advantage of preserving passivity by construction.

As mentioned above, having distributed linear elements presents a significant difficulty since the Telegrapher's equations that describe these elements do not have a direct representation in the time-domain in the form of differential equations shown in Equation (1.1). The key advantage of the Matrix-Padè approximation is that it can cast the Telegrapher's equations into a form that has time-domain representation similar to the ordinary differential equation of Equation (1.1). Applying MOR using Krylov-subspace techniques becomes therefore a straightforward task.

Although the Matrix-Padé techniques can be made to guarantee the passivity of the whole system, i.e., when the set of ordinary differential representation of the Telegrapher's equations is plugged into the rest of the circuit equations, the passivity of the reduced system after performing model-order reduction based on Krylov subspace methods is still in question.

One of the contributions in this thesis is the introduction of a closed form stamp for general transmission lines based on the Matrix-Padé approximation that can be used to represent a general linear network containing transmission lines in the form of a set of ordinary differential-algebraic equations in (1.1). The important issue with regard to this stamp is that applying MOR using Krylov subspace leads to a guaranteed passive reduced-order model.

Chapter 4 reviews the general techniques for simulating high-speed interconnects and introduces Matrix-Padé approximation. Chapter 5 introduces the stamp formulation based on the Matrix-Padé and presents a formal proof of the passivity of the reduced-order system produced using Krylov-subspace techniques.

1.3.5 The Arnoldi Process

The Arnoldi process is essentially an orthogonalization process that uses Modified Gram Schmidt [22] procedure to construct an orthonormal basis that spans the Euclidian space $\mathbb{R}^N$ of some vectors $\mathbf{m}_0, \mathbf{m}_1, \dots, \mathbf{m}_q \in \mathbb{R}^N$. However, the Arnoldi process [23] can be considered as a version of the Modified Gram Schmidt [22] process that is tuned to orthogonalize a set of vectors in which each vector is related to the previous vector through some matrix operator. In other words, if for a given set of vectors $\mathbf{m}_0, \mathbf{m}_1, \dots, \mathbf{m}_q \in \mathbb{R}^N$, it happens that $\mathbf{m}_q = \mathbf{A}\mathbf{m}_{q-1}, q > 0$, then the Modified Gram Schmidt [22] can be adapted such that the orthogonalization process is done much more accurately. This adapted process is what is usually referred to as the Arnoldi process.

The Arnoldi process is an essential step in performing model-reduction based on Krylov

subspace methods. The basic advantage of the Arnoldi process over classical Modified Gram Schmidt is that it enables the construction of the orthonormal basis without having the set of vectors $\mathbf{m}_0, \mathbf{m}_1, \dots, \mathbf{m}_q$ explicitly available. This in fact turns out to be a very desirable feature of the Arnoldi process for the Krylov model-reduction step. The reason for that is that the set of vectors that need to be orthogonalized for performing the model reduction step are related through a matrix that is not well conditioned. In practical terms, this means that higher-order vectors tend to be parallel and if computed explicitly and stored in a finite precision form then they become truly parallel. In fact, having them explicitly computed adds no more information to better the accuracy of the reduced model.

While the problem of constructing an accurate orthonormal basis for model reduction based on Krylov subspace methods has been addressed using the Arnoldi process in the context of Linear Time Invariant (LTI) systems, other types of systems possess certain features that make the Arnoldi process unapplicable. For example, in linear time variant systems, the set of vectors to be orthogonalized are related through a differential operator rather than an algebraic operator.

This thesis introduces a generalization of the Arnoldi process to perform orthogonalization on a set of vectors related through a differential operator. The proposed orthogonalization process has been used for the model reduction of linear time-varying systems in Chapter 7.

1.3.6 Steady-State Analysis of Nonlinear Circuits

While MOR techniques have had a successful record in the analysis and simulation of linear time invariant circuits and systems, little has been done to carry that success to nonlinear circuits. Nonetheless, analysis of nonlinear circuits is an indispensable tool in reducing the cost and time required to design new integrated circuits. This is due to the fact that computer simulation is capable of centering the design much faster than repeated fabrication and testing of new designs. However, one of the main difficulties in simulating general analog integrated

circuits is computing certain steady-state quantities such as harmonic distortion.

The problem of computing steady-state solution for a general circuit is basically that of finding the response of the circuit due to a periodically forcing input after the transient phase has died out. Although steady-state response can be computed via numerical integration of the set of differential equations describing the circuit voltages; such an approach can become very time consuming for circuits that have high quality or works in narrow-bands. In addition, quantities of interest to designers, such as third-order intermodulation and harmonic distortion, would be very inaccurate due to the numerical truncation error.

On the other hand, frequency domain simulation avoids these problems by converting the set of time-domain differential equations into a set of nonlinear algebraic equations that can be solved using iterative techniques such as *Newton-Raphson* (NR) method or continuation methods.

A frequency-domain approach tackles the problem of finding the steady-state by representing the circuit variables (e.g., voltages, currents, charges, and fluxes) using their Fourier series, in this formulation, so only periodic signals exist and transients are skipped.

Substituting this Fourier series representation into the set of differential equations naturally transforms them into a set of nonlinear algebraic equations. This approach is generically referred to as the *Harmonic Balance* (HB) approach, where various implementation techniques differ in the way of solving the resulting set of nonlinear algebraic equations.

Earlier approaches focused on using optimization as a vehicle to solve such a set of equations. The piecewise Harmonic Balance techniques [24], [25] is a good example of this category and is specifically geared towards networks that consist mainly of linear circuits with few nonlinear components. The efficiency of these technique lies in that they reduce the number of optimization variables to a minimum corresponding only to the terminal voltages (or currents) of the nonlinear elements. However, highly integrated circuits and *Monolithic Microwave Integrated Circuits* (MMIC's) tend to have a large number of nonlinear elements with each node

connected to linear as well as nonlinear devices. Under these circumstances, piecewise HB techniques tend to perform similar to classical optimization problems with huge number of optimization variables corresponding to the number of nodes (connected to nonlinear devices) multiplied by the number of harmonics (Fourier series coefficient) used to represent the voltage waveforms for each circuit variable.

Newton HB [26] has been presented in order to avoid problems associated with optimization problems. Newton HB is based on using the Newton-Raphson technique to solve the system of nonlinear algebraic equations, that arises in HB approach.

The Newton Raphson method seeks to find the solution of the nonlinear equations through successive updating of some initial guessed solution,

$$\mathbf{V}^{(j+1)} = \mathbf{V}^{(j)} - \delta \mathbf{V}^{(j)} \quad (1.5)$$

here $\mathbf{V}^{(j)}$ is the solution available from the previous iteration and $\delta \mathbf{V}^{(j)}$ is the update required to reach the actual solution and is given by:

$$\delta \mathbf{V}^{(j)} = \mathbf{J}(\mathbf{V}^{(j)})^{-1} \mathbf{F}(\mathbf{V}^{(j)}) \quad (1.6)$$

where $\mathbf{J}(\mathbf{V}^{(j)})$ is the Jacobian matrix and $\mathbf{F}(\mathbf{V}^{(j)})$ is the error in satisfying the system of equations, both computed at the present solution $\mathbf{V}^{(j)}$.

The advantage of using Newton over optimization HB is due to the fact that in optimization techniques the goal is always to solve the problem

$$\frac{d}{dx} (\mathbf{F}(x))^{T*} \mathbf{F}(x) = 0 \quad (1.7)$$

where instead of direct solution for $\mathbf{F}(x) = 0$. The difficulty in solving Equation (1.7) is that information about each of the individual contributors to the error in satisfying $\mathbf{F}(x)$ is lost when forming $\mathbf{F}^*(x)\mathbf{F}(x)$. However, as current design trends tend more towards larger level of integration with arbitrary strong nonlinear behavior and multi-tone excitation signals, the difficulties facing NR method becomes two-fold.

The first problem is related to computational considerations. Computing an update for the solution at the j^{th} iteration requires a linear system solution with the coefficient matrix $J(\mathbf{X}^{(j)})$. In general, this matrix is very large in size and exhibits a strong coupling between its elements in the presence of strong nonlinearity. Direct factorization of this would overwhelm even medium-sized computers. The second difficulty for a typical NR method is due to convergence issues. NR is locally convergent which means that convergence can be achieved only by finding an initial guess that is sufficiently close to the solution.

Continuation approaches [27] have been proposed to overcome the convergence problems associated with NR methods. In essence, a typical continuation method is a repeated solution for a large number of systems of nonlinear equations. These nonlinear systems of equations are basically obtained by transforming the set of original HB equations into another set of equations that has a trivial solution and then using some sweeping parameter to trace the solution trajectory back to the solution of the original systems.

Although continuation methods have proved to be efficient in overcoming the convergence problems of direct NR methods, they become computationally very expensive in the case of HB equations as they require the factorization of the Jacobian matrix for many values of the sweeping parameter.

To overcome the problems arising from the high computational cost associated with factorizing the Jacobian matrix, Krylov-subspace linear solvers have been introduced [28]. In this scheme, only the components of the Jacobian matrix need to be stored and a Krylov-subspace iterative solver is used. The main computational cost in each iteration is only due to the multiplication of the Jacobian matrix by a vector. The matrix-vector multiplication can be done efficiently through computing the effect of each component of the Jacobian matrix on the vector. However, efficiency of Krylov-subspace methods depends on the fast convergence of the iterative process. A crucial requirement for robust convergence is that the Jacobian matrix be well-conditioned, and hence using pre-conditioner is mandatory to deal with general matri-

ces. A pre-conditioner is an approximant matrix that approximates the inverse of the Jacobian matrix. Efficient pre-conditioning can be accomplished easily for circuits with mild nonlinearity using a block diagonal pre-conditioner matrix. Unfortunately, these simple block diagonal pre-conditioners become ineffective when the circuit contains strong nonlinearity.

This thesis provides another contribution that poses the problem from a totally different perspective. It describes a new algorithm based on a new concept of nonlinear circuit reduction. The crux of the proposed algorithm is to reduce the size of the Jacobian matrix quite drastically, yet it preserves the convergence robustness of continuation methods. This enables the original set of nonlinear equations to be replaced with a much smaller set of nonlinear equations.

Chapter 3 presents a background of the problem of steady-state analysis of nonlinear circuits in the frequency domain. Chapter 5 describes the proposed algorithm.

1.3.7 Model Reduction of Linear Periodically Time Varying Systems

Linear Periodically Time varying (LPTV) systems appear in different situations of circuit analysis. A typical example is the case of noise analysis for nonlinear systems, where the small signal approach is used. Another example is an RF component of a receiver subsystem. In these types of situations, the circuit under consideration behaves as a linear system w.r.t. the small signal, i.e., noise or RF sources. On the other hand, it exhibits a significant nonlinear behavior w.r.t. other signals, e.g., local oscillator signals. In such cases, the system appears to be linear for the small signal sources but with time-varying coefficients.

The problem of generating reduced order LPTV systems that capture the behavior of much larger LPTV systems has been addressed recently in [11], [12], [29], [30]. Although slightly different in mathematical formulation, both approaches are based on the concept of *Time-Varying systems* (TVS) function introduced by Zadeh [31], [32] to describe time-varying networks. However, this approach requires a solution of a large system of equations.

This thesis describes a new algorithm that does not require a solution of large system of equations. The proposed algorithm presents a two-fold (theoretical and computational) contribution. The theoretical aspect addresses the problem of approximating a TVS function that has its derivatives related recursively through a differential operator. This issue is discussed in Chapter 7.

On the computational side, the proposed algorithm does not require a huge linear system of equations to be solved. Instead, it takes advantage of the fact that periodically time-varying transfer functions have very short transient response in the right half-plane of the Laplace domain. This means that computing the time varying derivatives can be computed easily using transient time-domain integration for a short period of time. Chapter 7 describes LPTV systems and proposes a new algorithm for obtaining reduced order systems.

1.3.8 Simulation of Nonuniform Transmission Lines

As mentioned earlier, analysis of uniform transmission lines in the time-domain presents difficulties since these elements are naturally described in the frequency-domain through partial differential equations. Simulation of nonuniform transmission lines represents an even more difficult problem due to the fact that Per-Unit-Length (PUL) parameter matrices of the line become a function of the spatial variable.

This thesis presents a new algorithm based on the concept of Integrated Congruence Transform (ICT) [1]. The algorithm proposed in Chapter 8 casts the set of Telegraphers Equations into a set of algebraic relations that relate the voltages and currents at the terminals of the transmission lines conductors. This is done through projecting the differential operator of the Telegraphers Equations onto a much smaller Hilbert subspace spanned by the first few moments of the matrix-valued frequency-domain transfer function. The numerical efficiency of the proposed algorithm stems from the idea that the orthogonal basis for the space of the mo-

ments is generated without computing these moments explicitly.

1.4 Organization of the Thesis

Chapter 2 presents the Modified Nodal Analysis (MNA) representation of the circuit equations including linear, nonlinear and distributed elements. This formulation serves as the basis for mathematical development for the rest of the thesis. Chapter 3 introduces a background on the steady-state analysis techniques. Chapter 4 reviews simulation techniques for high-speed interconnect circuits. Chapter 5 presents the proposed algorithm for model reduction of high-speed interconnects. Chapter 6 presents the circuit reduction techniques for finding the steady-state solution of nonlinear circuits. Chapter 7 presents a new algorithm for model reduction of LPTV systems. Chapter 8 presents an algorithm for simulating nonuniform TL's based on Integrated Congruence Transform (ICT) through implicit moment calculation.

*

Chapter 2

Generalized Formulation of Circuit Equations

This chapter presents a general framework for the mathematical representation of general circuits. The formulation used in this chapter is general and will be used throughout the thesis in explaining the main objectives of circuit simulation. Section 2.1 presents the concept of Modified Nodal Admittance (MNA) for the formulation of linear circuits. Section 2.2 highlights the properties of the MNA matrices. Section 2.3 describes the extension of the MNA formulation to include circuits with nonlinear elements. Section 2.4 discusses the MNA formulation in the frequency-domain for circuits with MTL networks. Finally, Section 2.5 discusses the mixed frequency-time problem associated with transient simulation of MTL in the presence of nonlinear terminations.

2.1 MNA Formulation of Linear Circuits

Consider a circuit consisting of lumped RLC components and independent voltage and current sources. Using the MNA formulation [33] and [34], the circuit equations can be written in a

matrix form as:

$$C\dot{\mathbf{x}}(t) + G\mathbf{x}(t) = \mathbf{b}(t) \quad (2.1)$$

where

- $\mathbf{x}(t) \in \mathbb{R}^N$ is a vector of node voltage waveforms appended by independent voltage source and linear inductors' currents,
- $C \in \mathbb{R}^{N \times N}$ and $G \in \mathbb{R}^{N \times N}$ are matrices describing the linear lumped memory and memoryless elements, respectively,
- $\mathbf{b}(t)$ is a vector with entries determined by the independent voltage and current sources, and,
- N is the total number of variables, including node voltages, voltage sources, and linear inductor currents.

The G and C matrices are typically referred to as the MNA matrices. The advantage of using the MNA formulation is that the construction of the different terms in Equation (2.1) can be easily automated through using predefined stamps (or *stencils*) for each component. Figure 2.1 shows the stencils corresponding to the RLC lumped components in addition to independent current and voltage sources. Note that all what is needed to obtain the MNA formulation is the two terminal nodes of the element and its value.

2.2 Properties of the MNA Matrices

Consider the circuit of Figure 2.2. Using the MNA formulation of Figure 2.1 we get

Element	Circuit Schematic	Stamp
Resistor		$\begin{matrix} i \\ j \end{matrix} \begin{bmatrix} g & \cdots & -g \\ \vdots & \ddots & \vdots \\ -g & \cdots & g \end{bmatrix}$
Capacitor		$\begin{matrix} i \\ j \end{matrix} \begin{bmatrix} C & \cdots & -C \\ \vdots & \ddots & \vdots \\ -C & \cdots & C \end{bmatrix}$
Inductor		$\begin{matrix} i \\ j \end{matrix} \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \ddots & \vdots & 0 \\ 0 & \cdots & 0 & -1 \\ 1 & 0 & -1 & -sL \end{bmatrix}$
Voltage Source		$\begin{matrix} i \\ j \end{matrix} \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \ddots & \vdots & 0 \\ 0 & \cdots & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} E \end{bmatrix}$
Current Source		$\begin{matrix} i \\ j \end{matrix} \begin{bmatrix} J \\ \vdots \\ -J \end{bmatrix}$

Figure 2.1: Component Stamps in MNA formulation.

Figure 2.2: Illustration of the formation of the MNA matrices for a two-port RLC circuit example.

$$G = \begin{bmatrix} G_1 & -G_1 & 0 & 0 & 0 & 1 & 0 \\ -G_1 & G_1 + G_2 & -G_2 & 0 & 0 & 0 & 0 \\ 0 & -G_2 & G_2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & G_3 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 & & & \\ -1 & 0 & 0 & 0 & & 0 & \\ 0 & 0 & 0 & -1 & & & \end{bmatrix} \quad (2.2)$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & & & \\ 0 & C_c + C_1 & -C_c & 0 & & & \\ & & & & 0 & & \\ 0 & -C_c & C_2 + C_c & 0 & & & \\ 0 & 0 & 0 & 0 & & & \\ & & & & & L_1 & 0 & 0 \\ & & 0 & & & & 0 & 0 & 0 \\ & & & & & & & 0 & 0 & 0 \end{bmatrix} \quad (2.3)$$

$$\mathbf{b}(t) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \quad (2.4)$$

From equations (2.2) and (2.3) a general structure for $\mathbf{G}$, $\mathbf{C}$ and $\mathbf{x}(t)$ can be written as [15]:

$$\mathbf{C} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{N} & \mathbf{E} \\ -\mathbf{E}^T & \mathbf{0} \end{bmatrix}, \quad \mathbf{x}(t) = \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{i}(t) \end{bmatrix} \quad (2.5)$$

where $\mathbf{N}$, $\mathbf{Q}$ and $\mathbf{H}$ matrices contain the stamps for the resistors, capacitors, and inductors, respectively. The elements of $\mathbf{E}$ consists of ± 1 , such that $\mathbf{E}$ maps the current variables to the space of nodes in the network.

The matrices $\mathbf{Q}$, $\mathbf{N}$, and $\mathbf{H}$ are symmetric nonnegative definite provided that the original network is composed of passive linear elements, i.e., positive resistors, capacitors, and inductors only [15].

2.3 Nonlinear Elements

The inclusion of nonlinear elements in the circuit requires an additional term to be added to the equations in (2.1). A nonlinear element can be resistive, capacitive, or inductive. A circuit with nonlinear elements can be described by [34]:

$$\mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{G}\mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t)) = \mathbf{b}(t) \quad (2.6)$$

where $\mathbf{f}(\mathbf{x}(t))$ is a vector of nonlinear functions representing the contribution of nonlinear elements, and $\mathbf{x}(t)$ in this case is appended by the waveforms of the charge on the nonlinear capacitor and the flux on the nonlinear inductor.

As an example that illustrates this type of formulation that comprises nonlinear elements, consider the circuit shown in Figure 2.3. In this example we have:

Figure 2.3: A rectifier circuit with a nonlinear inductor.

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & g & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.8)$$

$$\mathbf{f}(\mathbf{x}(t)) = \begin{bmatrix} I_s(e^{\frac{v_1-v_2}{v_t}} - 1) \\ -I_s(e^{\frac{v_1-v_2}{v_t}} - 1) \\ 0 \\ -kv_2^2 \end{bmatrix} \quad (2.9)$$

$$\mathbf{x}(t) = [v_1(t) \ v_2(t) \ I_v(t) \ \phi(t)]^T \quad (2.10)$$

$$\mathbf{b}(t) = [0 \ 0 \ E(t) \ 0]^T \quad (2.11)$$

2.4 MNA Formulation for Transmission Lines

Transmission lines are generically referred to as interconnects. In general, interconnects exist at various levels of design hierarchy. Figure 2.4 shows that interconnects can exist at all hierarchy levels such as package structures, Multi-Chip Modules (MCM), Printed Circuit Boards (PCB), backplanes, and on-chip dies [4].

At lower frequencies, interconnects can be ignored and treated as simple short-circuits [3]. With the increasing quest for higher operating frequencies, however, interconnects do not behave as short-circuits, but rather as transmission lines with all phenomena associated with transmission lines such as delay, distortion, ringing and cross-talk [35]. This is due to the fact that, at higher frequencies, the electrical length of the interconnect becomes a significant fraction of the wavelength [36].

Under these conditions, distributed transmission line models based on per-unit-length parameters are needed to describe these effects accurately. Under a quasi Transverse Electromagnetic Mode (TEM) assumption [3], a lossy coupled multi-conductor transmission line (MTL),

Figure 2.4: High-Speed interconnects.

containing $m + 1$ conductors can be modelled by the Telegrapher's equations as [3]:

$$\frac{\partial \mathbf{v}(x, t)}{\partial x} = -\mathbf{R}\mathbf{i}(x, t) - \mathbf{L}\frac{\partial \mathbf{i}(x, t)}{\partial t} \quad (2.12a)$$

$$\frac{\partial \mathbf{i}(x, t)}{\partial x} = -\mathbf{G}\mathbf{v}(x, t) - \mathbf{C}\frac{\partial \mathbf{v}(x, t)}{\partial t} \quad (2.12b)$$

where $\mathbf{v}(x, t) \in \mathbb{R}^m$, and $\mathbf{i}(x, t) \in \mathbb{R}^m$ represent the voltages and currents, respectively, in the time-domain as functions of the displacement x along the line. $\mathbf{R}$, $\mathbf{L}$, $\mathbf{G}$, and $\mathbf{C} \in \mathbb{R}^{m \times m}$ are the per-unit-length (PUL) resistance, inductance, conductance and capacitance matrices, respectively. The $(m + 1)^{\text{th}}$ conductor here is assumed to be the reference conductor. Figure 2.5 presents a schematic illustration for an MTL. While the constituent equations of lumped models are described by simple algebraic equations, distributed elements are described by partial differential equations (2.12). This fact makes them difficult to be incorporated directly in the MNA formulation of Equation (2.1). Thus, having a stencil representing high-speed interconnects directly in the time-domain formulation of (2.1) is not a viable option. Nevertheless,

Figure 2.5: Schematic illustration for an MTL transmission line.

several approaches have been proposed in the literature to develop a stamp for transmission lines that can be used in the frequency-domain [37]. The following subsection presents an example of frequency-domain stamp developed based on the matrix exponential technique [38].

2.4.1 Matrix Exponential Transmission Line Stencil

Taking the Laplace transform of both sides of Equations (2.12) yields :

$$\frac{\partial}{\partial x} \mathbf{X}(x, s) = -(\mathbf{D} + s\mathbf{E})\mathbf{X}(x, s) \quad (2.13)$$

where

$$\mathbf{D} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{G} & \mathbf{0} \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \mathbf{0} & \mathbf{L} \\ \mathbf{C} & \mathbf{0} \end{bmatrix}, \quad \mathbf{X}(x, s) = \begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} \quad (2.14)$$

Equation (2.13) is simply a set of ordinary differential equations in the spatial variable x . A closed-form solution of (2.13) can be written in the form:

$$\begin{bmatrix} V(d, s) \\ I(d, s) \end{bmatrix} = e^{-(D+sE)d} \begin{bmatrix} V(0, s) \\ I(0, s) \end{bmatrix} \quad (2.15)$$

where d is the length of the line.

Equation (2.15) provides the hybrid parameters of the transmission line in the form of matrix exponential. Define

$$T(s) = \begin{bmatrix} T_{11}(s) & T_{12}(s) \\ T_{21}(s) & T_{22}(s) \end{bmatrix} \equiv e^{-(D+sE)d} \quad (2.16)$$

Using Equations (2.15) and (2.16), the relationship between the terminal voltages and currents of the transmission line can be written as:

$$A(s)V(s) + B(s)I(s) = 0 \quad (2.17)$$

where

$$V(s) = \begin{bmatrix} V(0, s) \\ V(d, s) \end{bmatrix}, \quad I(s) = \begin{bmatrix} I(0, s) \\ -I(d, s) \end{bmatrix} \quad (2.18)$$

The negative sign before the far end current in (2.18) is introduced to force the current at the far end to be flowing into the line as opposed to what is shown in Figure 2.5. $A(s)$ and $B(s)$ in (2.18) are given by :

$$A(s) = \begin{bmatrix} T_{11} & -U \\ T_{21} & 0 \end{bmatrix}, \quad B(s) = \begin{bmatrix} T_{12} & 0 \\ T_{22} & U \end{bmatrix}, \quad (2.19)$$

A frequency-domain stamp can be constructed by simply incorporating the algebraic relation of (2.17) into the MNA frequency-domain formulation of (2.1).

2.4.2 General MTL Formulation

This section describes the general MNA formulation for a general linear network consisting of lumped RLC components in addition to an arbitrary number of MTL circuits. Consider a linear network ϕ consisting of lumped RLC components and N_t lossy coupled MTL sets, with n_k coupled conductors in the k^{th} transmission line set. Assume the network ϕ has N_ϕ nodal variables. The MNA Equations in (2.1) can be written as

$$\begin{bmatrix} \mathbf{G}_\phi + s\mathbf{C}_\phi & \mathbf{D}_1 & \cdots & \mathbf{D}_{N_t} \\ \mathbf{A}_1(s)\mathbf{D}_1^T & \mathbf{B}_1(s) & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{A}_{N_t}(s)\mathbf{D}_{N_t}^T & \mathbf{0} & \mathbf{0} & \mathbf{B}_{N_t}(s) \end{bmatrix} \begin{bmatrix} \mathbf{X}_\phi(s) \\ \mathbf{I}_1(s) \\ \vdots \\ \mathbf{I}_{N_t}(s) \end{bmatrix} = \begin{bmatrix} \mathbf{B}_\phi(s) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (2.20)$$

where $\mathbf{D}_k = [d_{i,j}]$ is a selector matrix with elements $d_{i,j} \in \{0, 1\}$ where $i \in \{1, 2, \dots, N_\phi\}$, $j \in \{1, 2, \dots, 2n_k\}$ with a maximum of one non-zero element in each row or column, that maps $\mathbf{I}_k(s) \in \mathbb{R}^{2n_k}$, the vector of currents entering the subnetwork k , into the node space $\mathbb{R}^{N_\phi}$. Note that in this formulation, the terminal currents of the transmission line are appended to the set of unknowns in the MNA formulation. $\mathbf{G}_\phi$ and $\mathbf{C}_\phi$ in (2.20) contain the stamps of the lumped RLC components. $\mathbf{A}_k(s)$ and $\mathbf{B}_k(s)$ are given by (2.19).

2.4.3 An Example of MTL Formulation

As an example illustrating the basic idea of using the above analysis approach in obtaining an MNA formulation for a circuit containing a single MTL set, consider the circuit shown in Figure 2.6.

$$\begin{bmatrix} \mathbf{G}_\phi + s\mathbf{C}_\phi & \mathbf{D}_1 \\ \mathbf{A}_1(s)\mathbf{D}_1^T & \mathbf{B}_1(s) \end{bmatrix} \begin{bmatrix} \mathbf{X}_\phi \\ \mathbf{I}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_\phi \\ \mathbf{0} \end{bmatrix} \quad (2.21)$$

Figure 2.6: A circuit example containing a 2-Conductors transmission line.

$\mathbf{X}_\phi$ is the vector of unknowns in the standard MNA formulation as shown in Equation (2.1), such that

$$\mathbf{X}_\phi(s) = [V_1(s) \ V_2(s) \ V_3(s) \ V_4(s) \ V_5(s) \ I_E(s)]^T \quad (2.22)$$

$$\mathbf{B}_\phi = [0 \ 0 \ 0 \ 0 \ 0 \ E_i]^T \quad (2.23)$$

$$\mathbf{G}_\phi = \begin{bmatrix} g_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_1 & 0 & -g_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -g_1 & 0 & g_1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.24)$$

$$C_\phi = \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.25)$$

$$D_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.26)$$

and $A_1(s)$ and $B_1(s)$ are defined in (2.19).

2.5 Mixed Frequency-Time Problem

Section 2.4 has emphasized that representing MTL interconnects comes naturally in the frequency domain due to the distributed nature of these structures which is described by a set of partial differential equations. This pure frequency-domain nature of interconnects becomes a serious obstacle in simulating circuits containing nonlinear elements in addition to MTL networks, since nonlinear elements are only represented in the time-domain formulation. This problem of simulating high-speed interconnects terminated by nonlinear elements has been referred to in the literature as *Mixed Frequency-Time problem* [4], [39], [35]

Chapter 4 presents an overview for the various approaches that appeared in the literature to address this mixed frequency-time problem.

Chapter 3

Background on Steady-State Analysis

Techniques

Steady-state analysis is a crucial requirement in general commercial circuit simulators since it provides accurate insights into design metrics of interests to circuit designers. Examples of these metrics are intercept points and 1-dB gain compression point [5]. The basic idea underlying the concept of steady-state analysis techniques is that the response of a stable circuit due to a forcing periodic input must be periodic [40], [41]. Based on the knowledge that all the waveforms of the circuit variables are periodic, the problem can be formulated as a boundary-value problem [42]. Approaches for finding the steady-state operating point differ in the way this boundary-value problem is formulated. In general, these approaches are classified based on the domain in which the problem is formulated. Therefore, most of the techniques proposed in the literature may be classified under two categories: frequency- and time-domain based approaches. Other approaches that are based on using both domains have been explored more recently with the objective of combining the advantages of both approaches [43], [44], [45], [46].

This chapter presents a general overview for the various methods that appeared in the literature for finding the steady-state solution of nonlinear circuits. Section 3.1 describes the basic

idea common to those steady-state methods based on the time-domain approach. The Newton-Shooting method is described briefly in Section 3.1.2 as an example of this approach. Section 3.2 illustrates the Harmonic Balance method as an example of frequency-domain approaches.

3.1 Time-Domain Approaches

Time-domain approaches tackle the problem of computing the steady-state operating point as a boundary-value problem in the time-domain. The goal of a time-domain based technique, therefore, becomes that of finding a set of initial conditions for the circuit variables such that numerical solution of the MNA differential-algebraic equations (DAE) described in (2.6) results in the same initial starting point after integration for one oscillation period of the forcing input.

3.1.1 Conventional Transient Analysis

Transient analysis is a brute-force way to obtain the steady-state solution through approaching the problem as a classical initial-value problem. In initial-value problems, an initial condition for the circuit variables at $t = t_0$ is assumed to be known. The response is obtained by starting with this initial condition and stepping forward in time using numerical integration techniques until all transients have died out [33]. Fast Fourier transform (FFT) [47] is then applied to the signal obtained when the steady-state has been reached in order to obtain the frequency components of the circuit. Typical integration methods are based on Linear Multi-Step (LMS) methods such as the Trapezoidal and Backward Euler rules [48].

As an example, consider the Backward Euler technique, where the following formula is used to approximate the derivative of the circuit variables, $\mathbf{x}(t)$, at $t_1 = t_0 + h$,

$$\left. \frac{d\mathbf{x}(t)}{dt} \right|_{t=t_1} = \frac{\mathbf{x}(t_1) - \mathbf{x}(t_0)}{h} \quad (3.1)$$

Applying (3.1) to the MNA equations of (2.6) yields

$$\left(G + \frac{1}{h}C\right)x(t_1) + f(x(t_1)) = \frac{1}{h}Cx(t_0) + b(t_1) \quad (3.2)$$

Equation (3.2) can be written in the form

$$Ax(t_1) + f(x(t_1)) = u \quad (3.3)$$

where

$$A = G + \frac{1}{h}C; \quad u = \frac{1}{h}Cx(t_0) + b(t_1) \quad (3.4)$$

Thus, applying Backward Euler rule to (2.6), converts the set of DAE into a set of nonlinear algebraic equations with $x(t_1)$ as the set of unknowns. Solving a system of nonlinear equations can be attempted through one of the iterative methods such as Newton-Raphson (NR) method [49]. A typical NR method works iteratively to find the solution of (3.3) through proceeding with the iterations

$$x^{(i+1)}(t_1) = x^{(i)}(t_1) - [J(x^{(i)}(t_1))]^{-1} \psi(x^{(i)}(t_1)) \quad (3.5)$$

where $J(x)$ is the Jacobian matrix of (3.3) and is given by

$$J(x) = A + \frac{\partial f(x)}{\partial x} \quad (3.6)$$

and $\psi(x)$ is given by

$$\psi(x) \equiv Ax + f(x) - u \quad (3.7)$$

with $x^{(0)}(t_1)$ is an initial guess solution.

As shown from the above analysis, finding the solution at each time step requires several LU decompositions of the Jacobian matrix in order to solve the linear system of equations in (3.5). This presents a significant CPU cost for large circuits as the size of these matrices becomes typically in the order of thousands. The CPU cost becomes even more prohibitive in

circuits with high quality factor (high Q), where the circuit goes through many cycles before reaching the steady-state.

High CPU cost is not the only disadvantage of this approach. Other mechanisms in this approach introduce numerical errors that become a major source of inaccuracy. One of these mechanisms, in particular, is the truncation error (TE) associated with LMS integrations formulae. Another source of errors results from using FFT, where aliasing, interpolation and numerical noise due to TE degrade the accuracy of the frequency components [50]. Any failure to satisfy the periodicity condition, i.e., $\mathbf{x}(t) = \mathbf{x}(t + T)$, results in some form of noise floor that corrupts the accuracy of the frequency components [51].

Aliasing errors [17], [52] arise if the sampling period of the FFT is not less than half the period of the highest frequency in the signal. Thus, if the number of tones considered is not sufficient, the power of the high frequency tones that have been omitted is aliased in the low frequency tones causing further errors.

In addition, interpolation errors arise because FFT requires equally spaced time points while simulated data points do not satisfy this property in general. To reduce these FFT errors, tight error tolerance must be imposed on the DAE solver to take smaller time steps which results in a further large number of time steps causing an even higher CPU cost.

3.1.2 Shooting Method

The objective of shooting methods [40], [53], [54] is to accelerate the convergence of the transient time-domain analysis. In other words, shooting methods aim at skipping long transient cycles. The basic ideas in the shooting method is that at steady-state $\mathbf{x}(t) = \mathbf{x}(t + T)$ where T is the period of the forcing input and $\mathbf{x}(t)$ is the vector of waveforms of the circuit variables as defined in Chapter 2. Thus, if an initial condition vector $\mathbf{x}(0)$ can be found such that

$$\mathbf{x}(T) - \mathbf{x}(0) = \mathbf{0} \quad (3.8)$$

then the periodic solution can be reached in only one cycle of numerical integration. The question now becomes how to find such an initial condition that satisfies Equation (3.8). The condition set in (3.8) is a typical boundary-value problem and can be solved via the state-transition function [52] $\Phi(x(t), \tau)$, which is defined by

$$\Phi(x(t), \tau) = x(t + \tau) \quad (3.9)$$

In other words, a state transition function is a function that produces the state-variables of the system as an output using the value of the state variables at a different time point as an input and the difference in time between the two points. Thus,

$$x(T) = \Phi(x_0, T) \quad (3.10)$$

where $x_0 = x(0)$. Substituting from (3.10) in (3.8), we obtain

$$\Phi(x_0, T) - x_0 = 0 \quad (3.11)$$

which is a set of nonlinear equations in the initial condition x_0 . Thus, solving for the initial conditions, x_0 , can be done using iterative methods such as the NR method [49]. This method is typically referred to as Newton-Shooting technique [40], [41]. It is worth mentioning that other approaches based on gradient and extrapolation techniques have been proposed in the literature [54], [53]. This section describes the solution based on NR method as an example. NR starts with an initial guess for x_0 , say $x_0^{(0)}$, and then proceeds with the following iteration scheme

$$x_0^{(i+1)} = x_0^{(i)} - [J_s(x_0^{(i)})]^{-1} \Psi(x_0^{(i)}) \quad (3.12)$$

where $\Psi(x_0^{(i)})$ is the nonlinear function defined by

$$\Psi(x_0^{(i)}) \equiv \Phi(x_0^{(i)}, T) - x_0^{(i)} = 0, \quad (3.13)$$

$J_s(x_0^{(i)})$ is the Jacobian matrix of the shooting equations defined by

$$J_s(x_0^{(i)}) = \frac{\partial \Psi(x_0)}{\partial x_0} \Big|_{x_0=x_0^{(i)}} = \frac{\partial \Phi(x_0, T)}{\partial x_0} \Big|_{x_0=x_0^{(i)}} - U \quad (3.14)$$

U is an identity matrix and $\frac{\partial \Phi(\mathbf{x}_0, T)}{\partial \mathbf{x}_0}$ is the sensitivity matrix of the state transition w.r.t. the initial condition computed at $\mathbf{x}_0 = \mathbf{x}_0^{(i)}$.

Computing the sensitivity matrix requires working with the first variational equation which is derived by differentiating the MNA equations in (2.6) w.r.t. $\mathbf{x}_0$ to obtain

$$\mathbf{G} \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}_0} + \mathbf{C} \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}_0} + \frac{\partial \mathbf{f}(\mathbf{x}(t))}{\partial \mathbf{x}(t)} \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}_0} = \mathbf{0} \quad (3.15)$$

or

$$\mathbf{G}\mathbf{z}(t) + \mathbf{C}\dot{\mathbf{z}}(t) + \mathbf{A}(t)\mathbf{z}(t) = \mathbf{0} \quad (3.16)$$

where

$$\mathbf{z}(t) = \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}_0}; \quad \mathbf{A}(t) = \frac{\partial \mathbf{f}(\mathbf{x}(t))}{\partial \mathbf{x}(t)} \quad (3.17)$$

Note that $\mathbf{z}(t) \in \mathbb{R}^{N \times N}$ is a matrix with N representing the total number of variables in the MNA formulation. Using (3.10) we get

$$\mathbf{z}(t) = \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}_0} = \frac{\partial \Phi(\mathbf{x}_0, t)}{\partial \mathbf{x}_0} \quad (3.18)$$

or

$$\frac{\partial \Phi(\mathbf{x}_0, T)}{\partial \mathbf{x}_0} = \mathbf{z}(T) \quad (3.19)$$

and

$$\mathbf{z}(0) = \frac{\partial \mathbf{x}(0)}{\partial \mathbf{x}_0} = \frac{\partial \mathbf{x}(0)}{\partial \mathbf{x}(0)} = \mathbf{U} \quad (3.20)$$

The desired sensitivity matrix can, therefore, be computed by integrating the first variational equation in (3.16) starting from $t = 0$ up to $t = T$. Since $\mathbf{z}(t)$ is a matrix, this process has to be repeated N times for each column of the matrix $\mathbf{z}(t)$.

The major computational cost of the Newton-Shooting method comes from having to solve a system of linear equations at each iteration of the NR method in (3.12), with the matrix $\mathbf{J}_s$ as the coefficients matrix. This matrix is typically full and the CPU cost of solving this problem is $\mathcal{O}(N^3)$.

3.2 Harmonic Balance Approach

The Harmonic Balance (HB) [25], [24], [55], [56], [26] approach is a technique for finding the steady-state solution in the frequency-domain. The concept of the HB approach can be easily explained in the context of linear circuits as it provides more insights into the case for nonlinear circuits.

3.2.1 HB for Linear Circuits

Consider a linear circuit described by the MNA formulation

$$C \frac{dx}{dt} + Gx(t) = b(t) \quad (3.21)$$

where the C and G are the MNA matrices and $b(t)$ is a vector representing the contribution of the independent voltage and circuit sources. Assuming periodic input sources, $b(t)$ can be represented as a Fourier series summation:

$$b(t) = \sum_{k=-K}^K B_k e^{jk\omega t} \quad (3.22)$$

where ω is the fundamental frequency and B_k is a vector containing the k^{th} harmonic coefficient for each component in the vector $b(t)$.

Assuming a periodic response for $x(t)$,

$$x(t) = \sum_{k=-K}^K X_k e^{jk\omega t} \quad (3.23)$$

and substituting from (3.22) and (3.23) into (3.21), and making use of the orthogonality of the

Fourier series ¹ family results in,

$$\begin{bmatrix} Y(-K\omega) & & & \\ & \ddots & & \\ & & Y(0) & \\ & & & \ddots \\ & & & & Y(K\omega) \end{bmatrix} \begin{bmatrix} X_{-K} \\ \vdots \\ X_0 \\ \vdots \\ X_K \end{bmatrix} = \begin{bmatrix} B_{-K} \\ \vdots \\ B_0 \\ \vdots \\ B_K \end{bmatrix} \quad (3.24)$$

where

$$Y(k\omega) = G + jk\omega C \quad (3.25)$$

is the admittance matrix $Y(s)$ computed at $s = jk\omega$. Note that the matrix in (3.24) is block-diagonal. This means that solving (3.24) reduces to solving sets of decoupled systems of equations

$$Y(k\omega)X_k = B_k, \quad \text{where } k = -K, \dots, 0, \dots, K \quad (3.26)$$

3.2.2 HB for Nonlinear Circuits

Consider a nonlinear circuit described by the MNA equations

$$C\dot{x}(t) + Gx(t) + f(x(t)) = b(t) \quad (3.27)$$

In case of a periodic input, $b(t)$ and $x(t)$ are periodic in the steady state, i.e.,

$$b(t) = \sum_{k=-K}^K B_k e^{jk\omega t}, \quad x(t) = \sum_{k=-K}^K X_k e^{jk\omega t} \quad (3.28)$$

Consequently, $f(x(t))$ is also periodic and can be expressed as

$$f(x(t)) = \sum_{k=-K}^K F_k e^{jk\omega t}, \quad (3.29)$$

¹The orthogonality property for the Fourier series can be described by

$$\frac{1}{T} \int_0^T e^{jm\omega t} e^{jn\omega t} dt = \begin{cases} 0 & m \neq n \\ 1 & m = n \end{cases}$$

where $\mathbf{F}_k$ is a vector containing the k^{th} harmonic of each component in the vector $\mathbf{f}(\mathbf{x}(t))$. The total number of coefficients in the above Fourier series is given by

$$N_h = 2K + 1 \quad (3.30)$$

substituting from (3.28) and (3.29) into (3.27), and equating coefficient of the different harmonics, leads to a system of $N_{hb} = NN_h$ of nonlinear equations with the Fourier coefficients as the unknowns. The steady-state solution is therefore found by solving this set of nonlinear algebraic equations instead of having to deal with the original set of nonlinear DAE system.

3.2.3 An Example of Harmonic Balance Equations

A simple single node example of Figure 3.1 is considered to illustrate the HB formulation in the presence of nonlinear elements.

Figure 3.1: An example circuit for HB formulation.

The MNA equations for this circuit can be written as

$$\dot{x}(t) + x(t) + f(x(t)) = \frac{1}{2}A(e^{j\omega t} + e^{-j\omega t}) \quad (3.31)$$

where $f(x(t)) = I_s(e^{\frac{x(t)}{v_t}} - 1)$. In (3.31) $x(t)$ is the node voltage across the diode, and $f(x(t))$ represents the nonlinear current drawn by the diode. After the steady-state is reached, $x(t)$ can be expressed as

$$x(t) = \sum_{k=-K}^K X_k e^{jk\omega t} \quad (3.32)$$

The derivative of $x(t)$ can be written as

$$\dot{x}(t) = \sum_{k=-K}^K jk\omega X_k e^{jk\omega t} \quad (3.33)$$

and the nonlinear diode current can be expressed as

$$f(x(t)) = \sum_{k=-K}^K F_k e^{jk\omega t} \quad (3.34)$$

Plugging (3.32)- (3.34) into (3.31) we obtain

$$\sum_{k=-K}^K jk\omega X_k e^{jk\omega t} + \sum_{k=-K}^K X_k e^{jk\omega t} + \sum_{k=-K}^K F_k e^{jk\omega t} = \frac{1}{2}A(e^{j\omega t} + e^{-j\omega t}) \quad (3.35)$$

Equation (3.35) can be satisfied if and only if the coefficients of $e^{jk\omega t}$ are satisfied in both sides for $k = -K, \dots, K$. The following equations must therefore be satisfied

$$\begin{aligned} -jK\omega X_{-K} + X_{-K} + F_{-K} &= 0 \\ &\vdots \\ -j\omega X_{-1} + X_{-1} + F_{-1} &= \frac{1}{2}A \\ X_0 + F_0 &= 0 \\ j\omega X_1 + X_1 + F_1 &= \frac{1}{2}A \\ &\vdots \\ jK\omega X_K + X_K + F_K &= 0 \end{aligned} \quad (3.36)$$

3.2.4 General Formulation of the Harmonic Balance Equations

Based on the above illustration, a general formulation for the HB equations can be assembled in the following matrix form

$$\bar{G}X + j\Omega\bar{C}X + F(X) = B \quad (3.37)$$

where

$$\bar{G} = \begin{bmatrix} G & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & G & \\ & & & \ddots & 0 \\ 0 & \dots & 0 & G \end{bmatrix}; \quad \bar{C} = \begin{bmatrix} C & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & C & \\ & & & \ddots & 0 \\ 0 & \dots & 0 & C \end{bmatrix} \quad (3.38)$$

$$\Omega = \begin{bmatrix} -K\omega I & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & \\ & & & \ddots & 0 \\ 0 & \dots & 0 & K\omega I \end{bmatrix} \quad (3.39)$$

$$X = \begin{bmatrix} X_{-K}^T & \dots & X_0^T & \dots & X_K^T \end{bmatrix}^T \quad (3.40)$$

$$B = \begin{bmatrix} B_{-K}^T & \dots & B_0^T & \dots & B_K^T \end{bmatrix}^T \quad (3.41)$$

Here $\bar{G}$, $\bar{C}$, and $\Omega \in \mathbb{R}^{N_{hb}}$ are the matrices describing the linear part of the network.

3.2.5 Solving the Harmonic Balance Equations

Being a set of nonlinear equations with the harmonics of the circuit variables as the unknowns, HB equations can be solved using iterative methods such as NR method. NR method proceeds by starting with an initial guess $X^{(0)}$ and iterates through

$$X^{(j+1)} = X^{(j)} - [J_h(X^{(j)})]^{-1} \Psi_h(X^{(j)}) \quad (3.42)$$

where

$$\Psi_h(X) \equiv \bar{G}X + j\Omega\bar{C}X + F(X) - B \quad (3.43)$$

and $J_h(\mathbf{X}) \in \mathbb{C}^{N_h \times N_h}$ is the Jacobian matrix obtained using the following equation

$$\mathbf{J}_h(\mathbf{X}) \equiv \frac{\partial \Psi_h(\mathbf{X})}{\partial \mathbf{X}} = \bar{\mathbf{G}} + j\Omega\bar{\mathbf{C}} + \frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}} \quad (3.44)$$

The rest of this section presents further details regarding the evaluation of the different terms required in Equations (3.42)-(3.44) to perform the NR iterations.

Direct Fourier Transform

The direct Fourier transform (DFT) is the basic computational component common in computing all the terms needed in Equations (3.42)-(3.44). FFT is simply an efficient method to evaluate DFT when the number of time points in the waveform is a power of 2 [47]. Efficiency of the FFT method stems from the recursive structures that can be exploited to avoid matrix-vector products.

Although FFT and DFT are mathematically equivalent, the DFT approach has been selected for presentation in this section due to its simplicity. In addition, using DFT provides a mathematical concise formulation for explaining the evaluation of the Jacobian matrix $\mathbf{J}_h(\mathbf{X})$ and the nonlinear function vector $\mathbf{F}(\mathbf{X})$.

Let $x(t)$ be a periodic signal with a period $T = \frac{2\pi}{\omega}$, where $x(t)$ can be expressed as a Fourier series

$$x(t) = \sum_{k=-K}^K X_k e^{jk\omega t} \quad (3.45)$$

Let $x(t)$ be sampled at N_h equally spaced time points $[t_0, t_1, \dots, t_{N_h-1}]$ in the interval $[0, T]$ such that

$$t_n = n \frac{T}{N_h}, \quad n = 0, 1, \dots, N_h - 1 \quad (3.46)$$

where N_h is given by (3.30). Writing the Fourier series in (3.45) at each time point yields

$$x(t_0) = \sum_{k=-K}^K X_k e^{jk\omega t_0} \quad (3.47a)$$

$$x(t_1) = \sum_{k=-K}^K X_k e^{jk\omega t_1} \quad (3.47b)$$

$$\vdots$$

$$x(t_{N_h-1}) = \sum_{k=-K}^K X_k e^{jk\omega t_{N_h-1}} \quad (3.47c)$$

Equation (3.47) can therefore be assembled in a matrix form as follows

$$\begin{bmatrix} x_0 \\ \vdots \\ x_{N_h-1} \end{bmatrix} = \begin{bmatrix} e^{-jK\omega t_0} & \dots & 1 & \dots & e^{jK\omega t_0} \\ e^{-jK\omega t_1} & \dots & 1 & \dots & e^{jK\omega t_1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ e^{-jK\omega t_{N_h-2}} & \dots & 1 & \dots & e^{jK\omega t_{N_h-2}} \\ e^{-jK\omega t_{N_h-1}} & \dots & 1 & \dots & e^{jK\omega t_{N_h-1}} \end{bmatrix} \begin{bmatrix} X_{-K} \\ \vdots \\ X_0 \\ \vdots \\ X_K \end{bmatrix} \quad (3.48)$$

where $x_i = x(t_i)$ represents the time sample of $x(t)$ at time t_i . Equation (3.48) represents the inverse direct Fourier transform (IDFT) and can be put in a more compact representation as

$$\mathbf{X}_t = \mathbf{\Gamma}^{-1} \mathbf{X} \quad (3.49)$$

where $\mathbf{X}_t = [x_0 \ x_1 \ \dots \ x_{N_h-1}]$ represents the time samples, $\mathbf{X} = [X_{-K} \ \dots \ X_K]$ represents the harmonics, and $\mathbf{\Gamma}^{-1}$ is the IDFT operator.

Note that $\mathbf{\Gamma}^{-1}$ is independent of the frequency ω since the argument of the exponential function is independent of ω as can be shown from

$$k\omega t_i = k\left(\frac{2\pi}{T}\right)\left(i\left(\frac{T}{N_h}\right)\right) = ki\left(\frac{2\pi}{N_h}\right) \quad (3.50)$$

Thus, the only condition for $\mathbf{\Gamma}^{-1}$ to be independent of the frequency is that the time samples $[x_0 \ x_1 \ \dots \ x_{N_h-1}]$ be equally spaced over the period $[0, T]$. The DFT operator is thus

given by

$$\mathbf{X} = \Gamma \mathbf{X}_t \quad (3.51)$$

The main computational complexity of DFT (and IDFT) is equivalent to the cost associated of the classical matrix-vector multiplication, i.e., $\mathcal{O}(N_h^2)$. On the other hand, FFT (and IFFT) achieves exactly the same result but with computational complexity of $\mathcal{O}(N_h \log_2(N_h))$ via making use of the structure of Γ .

Computation of $F(\mathbf{X})$: Single-Variable Case

Computation of $F(\mathbf{X})$ is required for NR iterations as shown in equation (3.43). For simplicity of presentation, this section considers first the case where the circuit has only one variable, i.e., one waveform as illustrated by the circuit of Figure 3.1. The goal therefore becomes the evaluation of $F(\mathbf{X})$ given a solution vector $\mathbf{X} \in \mathbb{C}^{N_h}$.

Firstly, the time samples of the waveform $x(t)$ are obtained from $\mathbf{X}$ through using the IDFT operator,

$$\mathbf{X}_t = \Gamma^{-1} \mathbf{X} \quad (3.52)$$

where $\mathbf{X}_t = [x_0 \ x_1 \ \cdots \ x_{N_h-1}]$ is a vector containing the time samples of $x(t)$. Next, a vector, $\mathbf{F}_t$, is constructed by sampling $f(x(t))$ at the time points $\{t_0, t_1, \dots, t_{N_h-1}\}$, such that

$$\mathbf{F}_t = [f(x_0) \ f(x_1) \ \cdots \ f(x_{N_h-1})]^T \quad (3.53)$$

Finally, computing $F(\mathbf{X})$ can then be carried out using $\mathbf{F}_t$ and the DFT operator as follows

$$F(\mathbf{X}) = \Gamma \mathbf{F}_t \quad (3.54)$$

Thus, computing $F(\mathbf{X})$ may be expressed compactly as

$$F(\mathbf{X}) = \Gamma \bar{f}(\Gamma^{-1} \mathbf{X}) \quad (3.55)$$

where the mapping $\bar{f}(\cdot) : \mathbb{R}^{N_h} \rightarrow \mathbb{R}^{N_h}$ is defined by

$$\bar{f}([u_0 \ u_1 \ \dots \ u_{N_h-1}]^T) = [f(u_0) \ f(u_1) \ \dots \ f(u_{N_h-1})]^T \quad (3.56)$$

Computation of the Matrix $\partial F(X)/\partial X$: Single-Variable Case

Computation and inversion of the Jacobian matrix represent the main computational complexity (CPU time cost) in the HB approach. To illustrate the main concept without loss of generality, consider the case of the single-node (single-waveform) circuit shown in Figure 3.1.

From Equation (3.37) we have

$$J_h(X) \equiv \frac{\partial \Psi_h(X)}{\partial X} = \bar{G} + j\Omega\bar{C} + \frac{\partial F(X)}{\partial X} \quad (3.57)$$

where $\bar{G}$, and $\bar{C}$ are defined in equation (3.38), and Ω is defined in (3.39), and rewritten here for completeness

$$\bar{G} = \bar{C} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1 & \\ & & & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}; \quad \Omega = \begin{bmatrix} -K\omega & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & \\ & & & \ddots & 0 \\ 0 & \dots & 0 & K\omega \end{bmatrix} \quad (3.58)$$

The remaining term in Equation (3.57), $\frac{\partial F(X)}{\partial X}$, is now considered. Using the chain rule of differentiation we get

$$\frac{\partial F(X)}{\partial X} = \frac{\partial F(X)}{\partial X_t} \frac{\partial X_t}{\partial X} \quad (3.59)$$

where X_t is the vector of time samples previously defined. From (3.49) and (3.55) we get

$$\frac{\partial F(X)}{\partial X} = \Gamma \frac{\partial \bar{f}(X_t)}{\partial X_t} \Gamma^{-1} \quad (3.60)$$

where

$$\frac{\partial \bar{f}(\mathbf{X}_t)}{\partial \mathbf{X}_t} = \begin{bmatrix} \left. \frac{\partial f}{\partial x} \right|_{x=x(t_0)} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \left. \frac{\partial f}{\partial x} \right|_{x=x(t_{N_h-1})} \end{bmatrix} \quad (3.61)$$

At this point it is important to take note of the structure of $\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$. Figure 3.2 shows an approximate representation for the sparsity patterns of $\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$ for different degrees of nonlinearity. If $f(x)$ is a linear function of x (e.g., $f(x) = ax$, for some arbitrary constant a), then $\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$

Figure 3.2: Sparsity patterns of $\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$ for different degrees of nonlinearity (Single-Variable case).

reduces to

$$\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}} = a \mathbf{I}_{N_h} \quad (3.62)$$

where $\mathbf{I}_{N_h}$ is the identity matrix of size N_h . As shown in Figure 3.2, as the level of the nonlinearity increases, fill-ins start appearing in the matrix. These off-diagonal fill-ins indicate the presence of the coupling effects between the different frequency components.

Indexing Notation

A special notation is introduced in this section in order to facilitate explaining the evaluation of $F(\mathbf{X})$ and $\frac{\partial F(\mathbf{X})}{\partial \mathbf{X}}$ for circuits with multi-variables. Consider a circuit with N variables (voltages, currents, charges, or fluxes) excited by a periodic signal. Let the number of harmonics used to represent the waveforms of these variables at steady-state be K . Hence, the total number of unknowns, N_{hb} , is given by

$$N_{hb} = N(2K + 1) \quad (3.63)$$

Let $\mathbf{X} \in \mathbb{C}^{N_{hb}}$ be a vector of the unknowns. $\mathbf{X}$ is said to be in node-minor indexing mode if it is arranged as shown in Equation (3.40). Let node-major indexing mode denote the arrangement of the unknowns vector $\mathbf{X}$ such that

$$\mathbf{X}_{\text{major}} = \begin{bmatrix} \mathbf{X}_1^T & \cdots & \mathbf{X}_N^T \end{bmatrix}^T \quad (3.64)$$

where $\mathbf{X}_i \in \mathbb{C}^{2K+1}$ is a vector that contains *all* the harmonics of the i^{th} circuit variable.

Let P denote a permutation operator such that

$$\mathbf{X} = P\mathbf{X}_{\text{major}} \quad (3.65)$$

where $P \in \mathbb{R}^{N_{hb} \times N_{hb}}$ is an orthonormal matrix with elements $P_{ij} \in \{0, 1\}$. Thus, the node-major indexing scheme can be obtained from $\mathbf{X}$ through

$$\mathbf{X}_{\text{major}} = P^T \mathbf{X} \quad (3.66)$$

Evaluation of $F(\mathbf{X})$ and $\frac{\partial F(\mathbf{X})}{\partial \mathbf{X}}$: Multi-variable case

Consider a circuit with N variables (voltages, currents, charges, and fluxes). In this case, we have $\mathbf{f}(\mathbf{x}(t))$ in (2.6) can be written as

$$\mathbf{f}(\mathbf{x}(t)) = [f_1(\mathbf{x}(t)) \quad \cdots \quad f_N(\mathbf{x}(t))]^T \quad (3.67)$$

with $f_i(\mathbf{x}(t)) : \mathbb{R}^N \rightarrow \mathbb{R}$. Assume that this circuit is excited by a periodic signal and let the number of harmonics used to represent the waveforms of the circuit variables in steady-state be K . Define the mapping $\bar{\mathbf{f}}(\mathbf{u}) : \mathbb{R}^{N_{hb}} \rightarrow \mathbb{R}^{N_{hb}}$, as

$$\bar{\mathbf{f}}(\mathbf{u}) = \begin{bmatrix} \mathbf{f}(\mathbf{S}_0 \mathbf{u}) \\ \vdots \\ \mathbf{f}(\mathbf{S}_{N_h-1} \mathbf{u}) \end{bmatrix} \quad (3.68)$$

with $N_{hb} = N N_h$ and $N_h = 2K + 1$ is the number of time samples used to represent each waveform. In Equation (3.68) $\mathbf{u} \in \mathbb{R}^{N_{hb}}$ represents a vector of $2K + 1$ sample points for each one of the N circuit variables arranged in a node-major index mode, and $\mathbf{S}_i \in \mathbb{R}^{N \times N_{hb}}$ is a selector matrix that maps the i^{th} time sample point from each node to a single vector containing the i^{th} samples from all the nodes. Computing $\mathbf{F}(\mathbf{X})$ is therefore given by

$$\mathbf{F}(\mathbf{X}) = \mathbf{P}^T \Xi \bar{\mathbf{f}}(\Xi^{-1} \mathbf{P} \mathbf{X}) \quad (3.69)$$

where

$$\Xi^{-1} = \begin{bmatrix} \Gamma^{-1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \Gamma^{-1} & \\ & & \ddots & 0 \\ 0 & \dots & 0 & \Gamma^{-1} \end{bmatrix}; \quad \Xi = \begin{bmatrix} \Gamma & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \Gamma & \\ & & \ddots & 0 \\ 0 & \dots & 0 & \Gamma \end{bmatrix} \quad (3.70)$$

Computing the Jacobian term $\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$ follows directly from (3.68) and (3.69) as

$$\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}} = \mathbf{P}^T \Xi \mathcal{D} \bar{\mathbf{f}} \Xi^{-1} \mathbf{P} \quad (3.71)$$

where

$$\mathcal{D} \bar{\mathbf{f}} = \begin{bmatrix} \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{S}_0 \mathbf{u}} \mathbf{S}_0 \\ \vdots \\ \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{S}_{N_h-1} \mathbf{u}} \mathbf{S}_{N_h-1} \end{bmatrix} \quad (3.72)$$

where $u = \Xi^{-1}PX$.

Figure (3.3) shows a graphical representation for the sparsity patterns of the Jacobian matrix depending on the level of nonlinearity existing in the circuit. In the case of pure linear

Figure 3.3: Sparsity pattern of $\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$ for different degrees of nonlinearity (Multi-variables case).

circuits, the Jacobian matrix reduces to a block-diagonal matrix which was readily obtained from Section (3.2.1). For a circuit behaving strongly nonlinearly, the Jacobian matrix has coupled sparse blocks.

3.2.6 Iterative Methods

Sections 3.1 and 3.2 illustrated that the major bulk of computational complexity occurs during the iterative process of solving a system of nonlinear algebraic equations. At each step of this process a linear system of equations having the Jacobian of the nonlinear system as the coefficients matrix must be solved in order to update the solution.

For a circuit with N variables, this Jacobian matrix becomes a full matrix of size $N \times N$ in the case of shooting methods and a dense coupled $N_{hb} \times N_{hb}$ matrix in the case of the Harmonic Balance methods, where $N_{hb} = NN_h$, and N_h is the number of sample points in

each waveform.

Iterative approaches based on Krylov-subspace techniques [57] have been introduced in circuit simulation to overcome the bottleneck caused by the unwieldy structure of the Jacobian matrix in steady-state solution. The basic concept of using iterative Krylov-subspace methods is that the solution of the linear system

$$Ax = b \quad (3.73)$$

can be approximated through a small subspace, $Kr(A, b, k)$ defined by the column span of the vectors

$$Kr(A, b) = \text{ColumnSpan}[b \quad Ab \quad A^2b \quad \dots \quad A^{k-1}b] \quad (3.74)$$

There are various approaches to approximating the solution of (3.73) from $Kr(A, b, k)$ depending on the properties and nature of the matrix A . This section describes briefly the Generalized Minimal Residual (GMRES) method since it can be used with any general matrix.

The basic idea of the GMRES approach is to seek a solution that minimizes the norm of the residual error

$$r = \|b - Ax\| \quad (3.75)$$

The GMRES algorithm proceeds by choosing a solution, x_k , from $Kr(A, b, k)$ using

$$x_k = Q_k y_k \quad (3.76)$$

where Q_k is an orthogonal basis for $Kr(A, b, k)$, i.e.,

$$\text{ColumnSpan}[Q_k] = Kr(A, b, k) \quad (3.77)$$

$$Q_k^T Q_k = I_k \quad (k \times k \text{ identity matrix}) \quad (3.78)$$

The problem of finding x_k that minimizes (3.75) may now be described as

$$\min_{y_k} \|b - A Q_k y_k\| \quad (3.79)$$

Q_k can be constructed through the Arnoldi algorithm using b as a starting vector. Figure 3.4 displays a pseudo-code description for the Arnoldi algorithm. Since the main objective is *not*

Algorithm 3.2.1: ARNOLDI METHOD(A, b, k)

```

 $q_0 \leftarrow b/\|b\|$ 
for  $j \leftarrow 0$  to  $k - 1$ 
     $w_j = Aq_j$ 
    for  $i \leftarrow 0$  to  $j$ 
        do  $\left\{ \begin{array}{l} h_{i,j} \leftarrow q_i^T w_j \\ w_j \leftarrow w_j - h_{i,j} q_i \end{array} \right.$ 
     $h_{j+1,j} = \|w_j\|$ 
     $q_{j+1} = w_j/h_{j+1,j}$ 
 $Q_k \leftarrow [q_0 q_1 \cdots q_{k-1}]$ 

```

Figure 3.4: Pseudo-code description for the Arnoldi algorithm.

to construct the orthogonal basis Q_k but rather to find the minimizer of (3.79), one can make use of the following result from [57].

Theorem 3.2.1 *Let $A \in \mathbb{R}^{N \times N}$, $b \in \mathbb{R}^N$ and k be the inputs to the Arnoldi algorithm of Figure 3.4 and denote by Q_m the $N \times m$ matrix generated by running the algorithm for m , ($m \leq k$) steps and by $\bar{H}_m$ the $(m+1) \times m$ upper Hessenberg matrix whose nonzero entries are $h_{i,j}$, i.e.,*

$$\bar{H}_m = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,m-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,m-1} \\ 0 & h_{2,1} & h_{2,2} & \vdots \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & h_{m,m-1} \end{bmatrix} \quad (3.80)$$

Then

$$AQ_m = Q_{m+1}\bar{H}_m \quad (3.81)$$

By substituting from (3.81) into (3.79) and using the fact that the vector $b/\|b\|$ is the first column in the matrix Q_{k+1} , the approximate solution of (3.73), x_k , can be obtained from

$$\begin{aligned} x_k &= Q_k \times \arg y_k \min \|b - Q_{k+1}\bar{H}_m y_k\| \\ &= Q_k \times \arg y_k \min \|Q_{k+1}^T b - \bar{H}_m y_k\| \\ &= Q_k \times \arg y_k \min \|\beta e_1 - \bar{H}_m y_k\| \end{aligned} \quad (3.82)$$

where $\beta = \|b\|$. Finding the minimizer y_k in (3.82) is inexpensive since it requires the solution of a $(k+1) \times k$ least-squares problem where k is typically small.

GMRES algorithm thus keeps increasing k and checking for the accuracy of the corresponding solution through solving a least-squares minimization [22] problem.

Based on the above discussion of the GMRES algorithm, it can be generally stated that the main computational complexity of any general iterative solving techniques based on Krylov subspace is mainly associated with a matrix vector product (MVP) of the form

$$x^{(i+1)} = Ax^{(i)}, \quad i \geq 0 \quad (3.83)$$

followed by orthogonalizing $x^{(i+1)}$ with respect to $\{q^{(0)} \quad q^{(1)} \quad \dots \quad q^{(i)}\}$, i.e.,

$$x^{(i+1)} \leftarrow x^{(i+1)} - \sum_{j=0}^i \langle x^{(i+1)}, q^{(j)} \rangle q^{(j)} \quad (3.84)$$

Thus, using Krylov iterative techniques in solving for the steady-state solution avoids completely the matrix factorization problem and requires mainly MVP with the Jacobian matrix. This is an advantageous feature since the requirement that the Jacobian matrix be explicitly available is no longer needed. It is rather the effect of the Jacobian matrix as an operator on a vector as an operand that needs to be computed.

Unfortunately, these iterative linear solvers are not guaranteed to converge and require the use of a pre-conditioner to improve their convergence. Preconditioning the original system of (3.73) is carried by multiplying both sides of (3.73) by $\hat{A}^{-1}$. One then applies the iterative solver to the preconditioned system

$$\hat{A}^{-1}A^{-1}x = \hat{A}^{-1}b \quad (3.85)$$

The pre-conditioner $\hat{A}$ is chosen to be close to A that is also easy to convert. Generally, these are two conflicting goals. For mildly nonlinear problems, constructing $\hat{A}$ can be achieved by linearizing about the DC operating point and performing simple AC analysis at each $w_k = kw$. However, for strongly nonlinear problem this pre-conditioner is not sufficient [58], [59].

Applying Krylov iterative techniques in solving for the steady-state in the shooting method has been suggested in [60]. Techniques for using Krylov iterative techniques in solving HB problems have been proposed in [28], [61], [62], [63]. Another approach that combines the inexact Newton method [64] in solving nonlinear equations with Krylov-subspace iterative techniques was proposed in [65], [66].

3.3 Quasi-Periodic Steady-State Response

The approaches presented in this chapter thus far address the problem of finding steady-state solution under the assumption that the input signal contains one or more tones that are commensurate or harmonically related. Under this condition, the response waveforms are assumed to be periodic with harmonically related tones. General steady-state problems arise when the forcing input signal contains a number of non-harmonically related (non-commensurate) tones [67]. In this case, the waveforms of the circuit variables are said to be *quasi-periodic* or *almost periodic*. The notion of quasi-periodic signals can be formalized as follows [52]. A quasi-periodic waveform is waveform that can be uniformly approximated by the sum of a countable number

of sinusoidal waveforms. Hence $AP(\Lambda_K)$ denotes the space of all quasi-periodic signals if

$$\forall x(t) \in AP(\Lambda) \Rightarrow x(t) = \sum_{\omega_k \in \Lambda_K} X_k e^{j\omega_k t} \quad (3.86)$$

where

$$\Lambda_K = \{\omega_0, \omega_1, \omega_2, \dots, \omega_K\} \quad (3.87)$$

Evaluating quasi periodic steady-state response has been proposed through extending the HB approach explained in Section 3.2 using the Almost Periodic Transform (APFT) [68] and the Multi Dimensional Fourier Transform (MDFT) [69]. Other approaches that use mixed frequency-time scheme, to simulate almost periodic circuits, have been proposed in [44], [70], [68], [45], [71], [46].

Chapter 4

Simulation Techniques of High-Speed Interconnects

Chapter 2 emphasized the main difficulty in simulating high-speed distributed interconnect circuits. It was demonstrated that distributed networks are best described in the frequency-domain using the Telegrapher's equations. At further higher frequencies, the cross-sectional dimensions of the interconnects become of the same order as the wavelength (Figure 4.1). In this case, an interconnect structure must be treated as a dispersive structure and field components in the direction of propagation can no longer be neglected. Consequently, even the distributed models based on quasi-TEM approximations (e.g., Telegrapher's equations) become inaccurate in describing the interconnect behavior. In such situations, full-wave models [72], [73], [74] which take into account all possible field components, and satisfy all boundary conditions are required in order to give an accurate estimation of the high-frequency effects.

The techniques presented in this chapter and Chapters 5 and 8 cover quasi-TEM representation based on the Telegrapher's equations. Being partial-differential equations, Telegrapher's equations do not lend themselves naturally to transient time-domain simulation. This presents a significant difficulty for the analysis of circuits that have interconnect structures terminated

Figure 4.1: Interconnect cross-section and wavelength of the propagating signal.

by nonlinear elements since nonlinear elements are represented only in the time-domain. This problem was referred to in Section 2.5 as the mixed frequency-time problem.

This chapter presents a general review of the various approaches that appeared in literature to address this problem. These approaches can be broadly classified into two main categories: *Time-domain macromodelling* and *model-order reduction* (MOR) techniques. Section 4.1 describes briefly the issue of passivity and its importance in numerical transient time-domain simulation. Section 4.2 presents the macromodelling technique. The MOR approach is explained in Section 4.3.

4.1 Passivity Perseveration

Passivity implies that a network can not generate more energy than what it absorbs [75], [76]. Interconnect circuits and transmission lines are passive by nature. However, macromodels and reduced-order models representations of interconnects need not be passive.

Nonetheless, passivity is a crucial requirement that must be satisfied in any modelling approach for distributed networks. The reason behind that, as demonstrated in Figure 4.2, is that a non-passive macromodel (or reduced-order system) can lead to an unstable network when connected to passive terminations [77]. The instability here is merely a superficial phenomenon that only appears during the numerical simulation. Thus, the loss of passivity can introduce serious problems during transient simulation as numerical results can show artificial oscillations.

Figure 4.2: Passivity preservation of a macromodel connected to a passive termination

In mathematical terms, a network with an admittance matrix $\mathbf{Y}(s)$ is passive iff

1. $\mathbf{Y}(s^*) = \mathbf{Y}^*(s)$, where "*" is the complex conjugate operator.
2. $\mathbf{Y}(s)$ is a positive real matrix, i.e., $\mathbf{z}^{*T} [\mathbf{Y}^T(s^*) + \mathbf{Y}(s)] \mathbf{z} \geq 0$ for all complex values of s with $\Re(s) > 0$ and any arbitrary complex vector $\mathbf{z}$.

4.2 Time-Domain Macromodels

Time-domain macromodelling approaches, in general, tackle the problem of modelling mixed time-frequency problem via discretizing the transmission line into smaller parts that can be represented in the time-domain. The following subsections summarize a sample of the exiting discretization approaches for time-domain simulation.

4.2.1 RLC Lumped Networks

Representing high-speed distributed interconnects using lumped RLC sections can be considered as the simplest form of macromodelling approaches [3]. In fact, at fairly low frequencies simple RC trees (no floating capacitors and no resistors to ground) are enough to capture the delay introduced by transmission lines. For example, the work done in [78] provides a dominant time-constant approximation for monotonic step responses using RC trees.

With increasing operating frequencies, RC representation becomes inadequate to capture the ringing in the waveform and cascaded RLC sections need to be used. However, at even higher frequencies, lumped RLC sections are no longer accurate and a large number of RLC sections may be required giving rise to large networks that are CPU costly to simulate. Figure 4.3 depicts a graphical representation of the accuracy of different lumped components models versus the

Figure 4.3: Electrical interconnect models

4.2.2 Chebyshev Based Macromodels

In this macromodelling approach [14], spatial variations of multi-conductor transmission lines are expressed in terms of the Chebyshev orthonormal set of polynomials [79].

Rewriting the Telegrapher's equations in the Laplace-domain for an $(m + 1)$ -conductors transmission line gives the following form:

$$\frac{dV(x', s)}{dx'} = -(R(x') + sL(x'))I(x', s) \quad (4.1)$$

$$\frac{dI(x', s)}{dx'} = -(G(x') + sC(x'))V(x', s) \quad (4.2)$$

where, $V(x', s)$ and $I(x', s)$ are, respectively, the column vectors of the m lines voltages and currents. $R(x')$, $L(x')$, $C(x')$, and $G(x')$ are, respectively, frequency independent per-unit-length resistance, inductance, capacitance, and conductance $m \times m$ matrices, with s and x' are, respectively, the Laplace variable and distance across the line.

Development of the macromodel begins by expanding the variation of the voltages and currents across the line in terms of Chebyshev series, as shown in the following expressions:

$$V(x, s) = \sum_{m=0}^M V(x_m, s)g_m(x) \quad (4.3)$$

$$I(x, s) = \sum_{m=0}^M I(x_m, s)g_m(x) \quad (4.4)$$

where M is the number of polynomials used in the expressions, $g_m(x)$ is given by [79]:

$$g_m(x) = \frac{(1 - x^2)T_m(x)(-1)^{m+1}}{c_m M^2(x - x_m)} \quad (4.5)$$

and c_m is defined as:

$$c_m(x) = \begin{cases} 2, & m = 0, M \\ 1, & \text{otherwise.} \end{cases}$$

and $T_m = \cos(m \cos^{-1} x)$, here $x = 2x'/l - 1$ with l representing the length of the line.

Substituting from (4.3) and (4.4) into (4.1) and (4.2) and using certain properties of Chebyshev polynomials [79], it can be shown that (4.1) and (4.2) can be reduced to:

$$(\mathbf{A}^R + s\mathbf{A}^I)\mathbf{V}_t(s) + (\mathbf{B}^R + s\mathbf{B}^I)\mathbf{J}(s) = 0 \quad (4.6)$$

where

$$\mathbf{J}(s) = \begin{bmatrix} \mathbf{I}_t(s) \\ \hat{\mathbf{X}}(s) \end{bmatrix}$$

and $\mathbf{V}_t(s) = [\mathbf{V}_{\text{near}}^T(s) \quad \mathbf{V}_{\text{far}}^T(s)]^T$, $\mathbf{I}_t = [\mathbf{I}_{\text{near}}^T \quad \mathbf{I}_{\text{far}}^T]^T$ are, respectively, the vectors of terminal voltages and currents and $\hat{\mathbf{X}}(s)$ is a set of auxiliary variables. Expressions for $\mathbf{A}^R$, $\mathbf{A}^I$, $\mathbf{B}^R$, and $\mathbf{B}^I$ can be found in [14]. Thus, a time-domain description can be obtained directly as :

$$\mathbf{A}^R \begin{bmatrix} \mathbf{v}_{\text{near}}(t) \\ \mathbf{v}_{\text{far}}(t) \end{bmatrix} + \mathbf{A}^I \frac{d}{dt} \begin{bmatrix} \mathbf{v}_{\text{near}}(t) \\ \mathbf{v}_{\text{far}}(t) \end{bmatrix} + \mathbf{B}^R \begin{bmatrix} \mathbf{i}_{\text{near}}(t) \\ \mathbf{i}_{\text{far}}(t) \\ \hat{\mathbf{x}}(t) \end{bmatrix} + \mathbf{B}^I \frac{d}{dt} \begin{bmatrix} \mathbf{i}_{\text{near}}(t) \\ \mathbf{i}_{\text{far}}(t) \\ \hat{\mathbf{x}}(t) \end{bmatrix} = 0. \quad (4.7)$$

where $\mathbf{v}_{\text{near}}(t)$, $\mathbf{v}_{\text{far}}(t)$, $\mathbf{i}_{\text{near}}(t)$, $\mathbf{i}_{\text{far}}(t)$, and $\hat{\mathbf{x}}(t)$ are the inverse Laplace transform of $\mathbf{V}_{\text{near}}(s)$, $\mathbf{V}_{\text{far}}(s)$, $\mathbf{I}_{\text{near}}(s)$, $\mathbf{I}_{\text{far}}(s)$, and $\hat{\mathbf{X}}(s)$, respectively.

4.2.3 Matrix-Padé Macromodel

This macromodel was proposed in [16], [6] and later further improved in [80]. This section describes the basic idea of this macromodelling technique.

For uniform MTL, Equations (4.1) and (4.2) can be written in the closed form:

$$\begin{bmatrix} \mathbf{V}(d, s) \\ \mathbf{I}(d, s) \end{bmatrix} = e^{\mathbf{Z}} \begin{bmatrix} \mathbf{V}(0, s) \\ \mathbf{I}(0, s) \end{bmatrix} \quad (4.8)$$

where

$$\mathbf{Z} = (\mathbf{D} + s\mathbf{E})d, \quad \mathbf{D} = \begin{bmatrix} \mathbf{0} & -\mathbf{R} \\ -\mathbf{G} & \mathbf{0} \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \mathbf{0} & -\mathbf{L} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix} \quad (4.9)$$

and d is the length of the line. Note that in (4.9) R , G , L , and C are independent of the spatial distance x' and are considered as constants all over the line.

Derivation of the time-domain macromodel begins by using the matrix-Pad approximant of the exponential matrix $e^{\mathbf{Z}}$:

$$e^{\mathbf{Z}} \approx [\mathbf{P}_{N,M}(\mathbf{Z})]^{-1} \mathbf{Q}_{N,M}(\mathbf{Z}) \quad (4.10)$$

where $\mathbf{P}_{N,M}(\mathbf{Z})$ and $\mathbf{Q}_{N,M}(\mathbf{Z})$ are polynomial matrices expressed in terms of closed-form Pad rational functions as:

$$\begin{aligned} \mathbf{P}_{N,M}(\mathbf{Z}) &= \sum_{j=0}^N \frac{(M+N-j)!N!}{(M+N)!j!(M-j)!} (-\mathbf{Z})^j \\ \mathbf{Q}_{N,M}(\mathbf{Z}) &= \sum_{j=0}^M \frac{(M+N-j)!M!}{(M+N)!j!(N-j)!} \mathbf{Z}^j \end{aligned} \quad (4.11)$$

For $M = N = n$, the Matrix-Pad rational function of Equation (4.10) reduces to:

$$[\mathbf{P}_{n,n}(\mathbf{Z})]^{-1} \mathbf{Q}_{n,n}(\mathbf{Z}) = \prod_{i=1}^{n/2} [(a_i \mathbf{U} - \mathbf{Z})(a_i^* \mathbf{U} - \mathbf{Z})]^{-1} [(a_i \mathbf{U} + \mathbf{Z})(a_i^* \mathbf{U} + \mathbf{Z})] \quad (4.12)$$

for even values of n , and

$$\begin{aligned} [\mathbf{P}_{n,n}(\mathbf{Z})]^{-1} \mathbf{Q}_{n,n}(\mathbf{Z}) &= [\mathbf{P}_{n,n}(\mathbf{Z})]^{-1} \mathbf{Q}_{n,n}(\mathbf{Z}) \\ &= \prod_{i=1}^{(n-1)/2} [(a_i \mathbf{U} - \mathbf{Z})(a_i^* \mathbf{U} - \mathbf{Z})]^{-1} [(a_i \mathbf{U} + \mathbf{Z})(a_i^* \mathbf{U} + \mathbf{Z})] \end{aligned} \quad (4.13)$$

for odd values of n . Here, $\mathbf{U} \in \mathbb{R}^{2m \times 2m}$ is an identity matrix, $a_i = x_i + jy_i$, are the complex roots for $i > 0$, and a_0 is a real root. The symbol $*$ represents the complex conjugate operation. It is to be noted that the polynomial $\mathbf{P}_{N,M}(\mathbf{Z})$ is a strict Hurwitz polynomial [75]. This means that the real parts of the coefficients a_i and a_0 in (4.12) and (4.13) are positive.

Substituting from Equations (4.13) and (4.10) in (4.8)

$$\begin{aligned} \begin{bmatrix} V(d, s) \\ I(d, s) \end{bmatrix} &= [a_0 \mathbf{U} - \mathbf{Z}]^{-1} [a_0 \mathbf{U} + \mathbf{Z}] \cdots \\ &[(a_i \mathbf{U} - \mathbf{Z})(a_i^* \mathbf{U} - \mathbf{Z})]^{-1} [(a_i \mathbf{U} + \mathbf{Z})(a_i^* \mathbf{U} + \mathbf{Z})] \cdots \\ &[(a_{(n-1)/2} \mathbf{U} - \mathbf{Z})(a_{(n-1)/2}^* \mathbf{U} - \mathbf{Z})]^{-1} [(a_{(n-1)/2} \mathbf{U} + \mathbf{Z})(a_{(n-1)/2}^* \mathbf{U} + \mathbf{Z})] \begin{bmatrix} V(0, s) \\ I(0, s) \end{bmatrix} \end{aligned} \quad (4.14)$$

assuming odd values for n .

Thus, Equation (4.14) represents a mathematical formula that relates the voltages and currents at the terminal ends of the line. An alternative way to represent Equation (4.14) is through a cascaded connection of $(n-1)/2$ subsections as shown in Figure 4.4, where the terminal voltages and currents for the i^{th} subsection are related by:

$$[P_{n,n}(\mathbf{Z})]_{i+1} \begin{bmatrix} V_{i+1} \\ I_{i+1} \end{bmatrix} = [Q_{n,n}(\mathbf{Z})]_i \begin{bmatrix} V_i \\ I_i \end{bmatrix} \quad (4.15)$$

and the matrices $[P_{n,n}(\mathbf{Z})]_i$ and $[Q_{n,n}(\mathbf{Z})]_i$ are expressed by the following equations:

$$\begin{aligned} [P_{n,n}(\mathbf{Z})]_0 &= \begin{bmatrix} a_0 \mathbf{U} & (\mathbf{L}s + \mathbf{R})d \\ (\mathbf{C}s + \mathbf{G})d & a_0 \mathbf{U} \end{bmatrix} \\ [Q_{n,n}(\mathbf{Z})]_0 &= \begin{bmatrix} a_0 \mathbf{U} & -(\mathbf{L}s + \mathbf{R})d \\ -(\mathbf{C}s + \mathbf{G})d & a_0 \mathbf{U} \end{bmatrix} \end{aligned} \quad (4.16)$$

for $i = 0$, and

$$\begin{aligned} [P_{n,n}(\mathbf{Z})]_i &= \begin{bmatrix} \Psi_i & 2x_i(\mathbf{L}s + \mathbf{R})d \\ 2x_i(\mathbf{C}s + \mathbf{G})d & \Psi_i \end{bmatrix} \\ [Q_{n,n}(\mathbf{Z})]_i &= \begin{bmatrix} \Psi_i & -2x_i(\mathbf{L}s + \mathbf{R})d \\ -2x_i(\mathbf{C}s + \mathbf{G})d & \Psi_i \end{bmatrix} \end{aligned} \quad (4.17)$$

Figure 4.4: Multiconductor transmission line represented by subsections of Padé model.

for $i > 0$, where

$$\Psi_i = LCd^2s^2 + (LG + RC)d^2s + RGd^2 + (x_i^2 + y_i^2)U \quad (4.18)$$

A time-domain representation for the i^{th} subsection can thus be given by:

$$\begin{bmatrix} [p_a]_i & [q_a]_i \end{bmatrix} \begin{bmatrix} \dot{x}_{i+1} \\ \dot{x}_i \end{bmatrix} + \begin{bmatrix} [p_b]_i & [q_b]_i \end{bmatrix} \begin{bmatrix} x_{i+1} \\ x_i \end{bmatrix} = 0 \quad (4.19)$$

where the variables of (4.19) are computed using (4.12)-(4.19) as follows. For the second-order subsections described by (4.17) and (4.18), we have

$$\begin{aligned} \begin{bmatrix} p_a \end{bmatrix}_i &= \begin{bmatrix} U_X \\ 0 \\ M_i \end{bmatrix}, & \begin{bmatrix} p_b \end{bmatrix}_i &= \begin{bmatrix} U_Y \\ 0 \\ S_i \end{bmatrix} \\ \begin{bmatrix} q_a \end{bmatrix}_i &= \begin{bmatrix} 0 \\ U_X \\ N_i \end{bmatrix}, & \begin{bmatrix} q_b \end{bmatrix}_i &= \begin{bmatrix} 0 \\ U_Y \\ T_i \end{bmatrix} \\ \mathbf{x}_{i+1} &= \begin{bmatrix} v_{i+1}^T & i_{i+1}^T & \gamma_{i+1}^T & \rho_{i+1}^T \end{bmatrix}^T, & \mathbf{x}_i &= \begin{bmatrix} v_i^T & i_i^T & \gamma_i^T & \rho_i^T \end{bmatrix}^T \end{aligned} \quad (4.20)$$

where γ_i, ρ_i are extra state variables needed for the realization of each of the complex poles

and

$$\begin{aligned}
 M_i &= \begin{bmatrix} (LG + RC)d^2 & 2x_i Ld & LCd^2 & 0 \\ 2x_i Cd & (LG + RC)d^2 & 0 & LCd^2 \end{bmatrix} \\
 N_i &= \begin{bmatrix} -(LG + RC)d^2 & 2x_i Ld & -LCd^2 & 0 \\ 2x_i Cd & -(LG + RC)d^2 & 0 & -LCd^2 \end{bmatrix} \\
 S_i &= \begin{bmatrix} RGd^2 + (x_i^2 + y_i^2)U & 2x_i Rd & 0 & 0 \\ 2x_i Gd & RGd^2 + (x_i^2 + y_i^2)U & 0 & 0 \end{bmatrix} \\
 T_i &= \begin{bmatrix} -(RGd^2 + (x_i^2 + y_i^2)U) & 2x_i Rd & 0 & 0 \\ 2x_i Gd & -(RGd^2 + (x_i^2 + y_i^2)U) & 0 & 0 \end{bmatrix} \\
 U_X &= \begin{bmatrix} U & 0 & 0 & 0 \\ 0 & U & 0 & 0 \end{bmatrix} \\
 U_Y &= \begin{bmatrix} 0 & 0 & -U & 0 \\ 0 & 0 & 0 & -U \end{bmatrix}
 \end{aligned} \tag{4.21}$$

For the first-order subsection described by (4.15) and (4.17), we have

$$\begin{aligned}
 \begin{bmatrix} p_a \end{bmatrix}_0 &= \begin{bmatrix} M_0 \end{bmatrix} = \begin{bmatrix} 0 & Ld & 0 & 0 \\ Cd & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} p_b \end{bmatrix}_0 = \begin{bmatrix} S_0 \end{bmatrix} = \begin{bmatrix} a_0 U & Rd & 0 & 0 \\ Gd & a_0 U & 0 & 0 \end{bmatrix} \\
 \begin{bmatrix} q_a \end{bmatrix}_0 &= \begin{bmatrix} N_0 \end{bmatrix} = \begin{bmatrix} 0 & Ld \\ Cd & 0 \end{bmatrix}, \quad \begin{bmatrix} q_b \end{bmatrix}_0 = \begin{bmatrix} T_0 \end{bmatrix} = \begin{bmatrix} -a_0 U & Rd \\ Gd & -a_0 U \end{bmatrix} \\
 \mathbf{x}_1 &= \begin{bmatrix} v_1^T & i_1^T & \gamma_1^T & \rho_1^T \end{bmatrix}^T, \quad \mathbf{x}_0 = \begin{bmatrix} v_0^T & i_0^T \end{bmatrix}^T
 \end{aligned} \tag{4.22}$$

Assembling differential equations from each subsection, a time-domain stamp can be written for the whole transmission line in the form

$$\Psi \frac{\partial}{\partial t} \mathbf{x}(t) + \Phi \mathbf{x}(t) = 0 \tag{4.23}$$

where $x(t) = \begin{bmatrix} x_0^T & x_1^T & x_2^T & \cdots & x_{\theta-1}^T & x_\theta^T & x_{\theta+1}^T \end{bmatrix}^T$, and

$$\theta = \begin{cases} (n-1)/2 & n \text{ is odd} \\ n/2 & n \text{ is even} \end{cases}$$

$$\Psi = \begin{bmatrix} U_X & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & U_X & 0 & \cdots & 0 & 0 & 0 \\ M_\theta & N_\theta & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & U_X & \cdots & 0 & 0 & 0 \\ 0 & M_{\theta-1} & N_{\theta-1} & & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & U_X & 0 \\ 0 & 0 & 0 & \cdots & M_1 & N_1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & M_0 & N_0 \end{bmatrix} \quad (4.24)$$

$$\Phi = \begin{bmatrix} U_Y & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & U_Y & 0 & \cdots & 0 & 0 & 0 \\ S_\theta & T_\theta & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & U_Y & \cdots & 0 & 0 & 0 \\ 0 & S_{\theta-1} & T_{\theta-1} & & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & U_Y & 0 \\ 0 & 0 & 0 & \cdots & S_1 & T_1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & S_0 & T_0 \end{bmatrix} \quad (4.25)$$

4.3 Model-Order Reduction

A typical model-order reduction (MOR) technique would take, as an input, a system represented by large matrices constructed from stamps of lumped and distributed components and produces, as an output, another system represented by much smaller matrices. The underlying concept of MOR is that a large matrix can be approximated by its dominant eigenspace where a dominant eigenspace here refers to the space of the eigenvectors associated with the dominant eigenvalues [81], [82], [83]. MOR techniques can be classified under two main categories: explicit moment matching based on Padé approximants, and implicit moment matching based on projecting the large matrix on its dominant eigenspace.

4.3.1 Explicit Moment Matching

Explicit moment matching (EMM) techniques [38], [19], [84], [2] work by constructing a direct Padé approximant for the impulse response of the network in the frequency-domain. To illustrate this concept, consider a single-input single-output linear network with a frequency response $h(s) \in \mathbb{C}$. Using a Taylor's expansion around some point in the frequency domain [19] s_0 , $h(s)$ can be expressed as:

$$h(s) = m_0 + m_1(s - s_0) + m_2(s - s_0)^2 + \dots \quad (4.26)$$

where $m_i, i = \{0, 1, \dots\}$ is the i^{th} moment (or scaled derivative) of $h(s)$ w.r.t. s at $s = s_0$.

EMM techniques use the idea of Padé approximant to construct a reduced-order description for $h(s)$. Padé approximants are basically rational function approximations of the form $q_L(s)/q_M(s)$ where $q_L(s)$ and $q_M(s)$ are two polynomials of order L and M , respectively. To find such an approximation we write $h(s)$ as

$$\begin{aligned} h(s) &= \frac{a_0 + a_1(s - s_0) + \dots + a_L(s - s_0)^L}{1 + b_1(s - s_0) + \dots + b_M(s - s_0)^M} \\ &= m_0 + m_1(s - s_0) + m_2(s - s_0)^2 + \dots \end{aligned} \quad (4.27)$$

Cross-multiplying Equation (4.27) and equating similar powers of s , coefficients a_i and b_i are obtained as

$$\begin{bmatrix} m_{L-M+1} & m_{L-M+2} & \cdots & m_L \\ m_{L-M+2} & \ddots & \ddots & m_{L+1} \\ \vdots & \ddots & \ddots & \vdots \\ m_L & m_{L-M+1} & \cdots & m_{L+M-1} \end{bmatrix} \begin{bmatrix} b_M \\ b_{M-1} \\ \vdots \\ b_1 \end{bmatrix} = - \begin{bmatrix} m_{L+1} \\ m_{L+2} \\ \vdots \\ m_{L+M} \end{bmatrix} \quad (4.28)$$

$$\begin{aligned} a_0 &= m_0 \\ a_1 &= m_1 + b_1 m_0 \\ a_2 &= m_2 + b_1 m_1 + b_2 m_0 \\ &\vdots \\ a_L &= m_L + \sum_{i=1}^{\min(L,M)} b_i m_{L-i} \end{aligned} \quad (4.29)$$

Thus, obtaining Padé-approximant is a simple procedure once the moments of the transfer function are obtained.

Computing the moments of the transfer function can be computed with low CPU cost and it can be shown that it is equivalent to finding the DC operating point of the network [19]. To explain how to obtain the moments, consider a linear multiport network described by the MNA equations as shown in Equation (2.1)

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C}\dot{\mathbf{x}}(t) = \mathbf{B}\mathbf{u}(t), \quad \mathbf{i}(t) = \mathbf{B}^T \mathbf{x}(t) \quad (4.30)$$

where $\mathbf{u}(t)$ and $\mathbf{i}(t) \in \mathbb{R}_p^N$ represent the input and output of the networks. $\mathbf{G}$ and $\mathbf{C} \in \mathbb{R}^{N \times N}$ represent the memoryless and memory matrices in the network as explained in Section 2.1. $\mathbf{B} \in \mathbb{R}^{N \times N_p}$ is a selector matrix that maps the input ports to the rest of the network nodes, and N_p and N are the number of ports and number of variables in the network, respectively. Equation (4.30) can be written in the frequency-domain as

$$(\mathbf{I}_N - s\mathbf{A})\mathbf{X}(s) = \mathbf{R}\mathbf{U}(s), \quad \mathbf{I}(s) = \mathbf{B}^T \mathbf{X}(s) \quad (4.31)$$

where

$$A = -G^{-1}C \quad \text{and} \quad R = G^{-1}B \quad (4.32)$$

To obtain the impulse response, we substitute for $U(s)$ with the Laplace-domain function of the dirac-delta function, i.e., $U(s) = 1$ [85]. Hence, the Laplace-domain impulse response can be written as

$$\begin{aligned} H(s) &= (I_N - sA)^{-1}R \\ &= M_0 + M_1s + M_2s^2 + \dots \end{aligned} \quad (4.33)$$

where $M_0, M_1, \dots, M_i, \dots$ are the block moments of $H(s)$ w.r.t. s .

From Equation (4.33) it can be shown that

$$\begin{aligned} M_0 &= R \\ M_i &= AM_{i-1} = A^i M_0, \quad i > 0 \end{aligned} \quad (4.34)$$

EMM techniques generally face two main difficulties. The first difficulty is numerical and is directly related to the explicit computation of the moments in (4.34). In fact, experience shows that the matrix in (4.28) becomes ill-conditioned above a size of 10. This means that practically no further information can be obtained from using higher order moments in the Padé approximant. This problem has been addressed through various heuristic approaches. For example, the Complex Frequency Hopping (CFH) algorithm in [38] makes use of multiple expansion points in order to construct a unified rational approximation that extends the matching range in the frequency-domain beyond what can be obtained through a single expansion point.

The second difficulty is passivity. As mentioned in Section 4.1 a macromodel representing interconnects should be passive. Unfortunately, in EMM techniques, there is no absolute guarantee that the resulting reduced-order model is passive.

4.3.2 Implicit Moment Matching

Techniques of Implicit Moment Matching (IMM) are based on Krylov subspace approaches which were introduced by Nikolai Krylov in 1931 [86], [87]. These approaches have been recently carried over to circuit simulation as a way to perform MOR on high-speed interconnects [88], [81], [15], [89], [77], [83], [90], [18].

To explain the basic concept of MOR based on Krylov subspace techniques, consider a lumped multiport network whose impulse response is described using the MNA formulation

$$(G + sC) H(s) = B \quad (4.35)$$

or

$$(I - sA) H(s) = R \quad (4.36)$$

where

$$A = -G^{-1}C \quad R = G^{-1}B \quad (4.37)$$

The main goal of a typical Krylov-subspace procedure is to approximate the matrix A in (4.36) with a reduced-order matrix that preserves its leading eigenvalues, i.e., those eigenvalues with the largest absolute values. Matching the moments of the transfer function comes as a by-product of this approximation step. The approximation process is accomplished via projecting A on a subspace corresponding to the dominant eigenspace.

In general, the dominant eigenspace of the matrix A can be obtained from the column space of the matrix K defined by

$$K = \begin{bmatrix} R & AR & A^2R & \cdots & A^qR \end{bmatrix} \quad (4.38)$$

or using (4.34)

$$K = \begin{bmatrix} M_0 & M_1 & \cdots & M_q \end{bmatrix} \quad (4.39)$$

where $K \in \mathbb{R}^{N \times k}$ is a rectangular matrix with $k = qN_p$.

A typical Krylov-subspace method in its simplest form would proceed by constructing an orthonormal basis, Q , that spans the column space of K . Obtaining a reduced-order approximant, $\hat{A} \in \mathbb{R}^{k \times k}$, for A may then be obtained simply via the congruent transform, $\hat{A} = Q^T A Q$, with the approximation being efficient if $k \ll N$.

There are two critical issues that need to be stressed at this point. The first issue is related to the numerical aspect of Krylov-subspace methods in general. It has been shown in the literature [81] that the columns of the matrix K tend to align very quickly with the eigenvector corresponding to the largest (in the absolute sense) eigenvalue. However, in finite precision computations, columns of K seem to be parallel and adding high order moments to K does not further the accuracy of the reduced-order model. In fact, this is the main reason behind the numerical difficulties in EMM techniques mentioned in the previous subsection. On the other hand, the key feature of IMM based on Krylov-subspace that provides the numerical edge over EMM techniques, is its ability to construct the orthonormal basis Q without having to compute K explicitly. The algorithm that achieves that is called *the Arnoldi process* [91], [57] and is explained in the pseudo-code procedure of Figure 4.5. The fundamental idea in the above mentioned algorithm is that the moments are related through an algebraic operator and a conventional orthogonalization procedure can take advantage of that to avoid computing the moments explicitly.

The second critical issue that needs to be stressed here is the passivity of the reduced-order model. While Krylov-subspace methods can provide a numerical advantage over EMM techniques, the issue of guaranteeing the passivity of the reduced model remained an unsolved problem until 1997 when the PRIMA algorithm [15] was introduced. The PRIMA algorithm showed that if the network to be reduced has its matrices G , C , and B formulated as shown in Equation (2.5), then the reduced system described by:

$$(\hat{G} + s\hat{C})\hat{H}(s) = \hat{B}\hat{U}(s) \quad (4.40)$$

Algorithm 4.3.1: ARNOLDI METHOD(G, C, R, q)

```

 $[q_0, \mathcal{R}] \leftarrow qr(R)$   comment:  $qr$  factorization of  $R$ 

for  $j \leftarrow 1$  to  $q$ 
     $z \leftarrow G^{-1} C q_{j-1}$ 
    for  $i \leftarrow 0$  to  $j - 1$ 
        do  $\left\{ \begin{array}{l} h \leftarrow q_i^T z \\ z \leftarrow z - q_i h \end{array} \right.$ 
     $[q_j, \mathcal{R}] \leftarrow qr(z)$   comment:  $qr$  factorization of  $z$ 

 $Q \leftarrow [q_0 \ q_1 \ \cdots \ q_q]$ 
return ( $Q$ )

```

Figure 4.5: Pseudo-code description of Block Arnoldi process.

where

$$\hat{G} = Q^T G Q, \quad \hat{C} = Q^T C Q, \quad \text{and} \quad \hat{B} = Q^T B \quad (4.41)$$

is guaranteed to be passive.

As shown in Equation (4.41), the key in achieving the passivity by construction in the reduced-order system is the reduction of both the G and C matrices simultaneously through congruent transformation.

4.3.3 MOR Based on Integrated Congruence Transform

The basic concept in this technique of MOR has general similarities in the approach to the idea of IMM techniques described in Section 4.3.2.

This approach, however, does not require macromodelling techniques of section 4.2 to cast the Telegrapher's partial differential equations into ordinary differential equations. It rather applies model-order reduction directly to the Telegrapher's equations [1], [92].

• **Step 1: Basis Construction**

In order to explain the algorithm, the Telegrapher's equations in (2.12) are rewritten in the Laplace-domain as

$$T \frac{d\mathbf{X}(z, s)}{dz} = -(N(z) + sM(z))\mathbf{X}(z, s) \quad (4.42)$$

where

$$\begin{aligned} N(z) &= \begin{bmatrix} \mathbf{R}(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{G}(z) \end{bmatrix}, \quad M(z) = \begin{bmatrix} \mathbf{L}(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(z) \end{bmatrix} \\ T &= \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \\ \mathbf{I}_m & \mathbf{0} \end{bmatrix}, \quad \mathbf{X}(z, s) = \begin{bmatrix} \mathbf{I}(z, s) \\ \mathbf{V}(z, s) \end{bmatrix} \end{aligned} \quad (4.43)$$

and the spatial variable x in Equation (2.12) has been replaced by z for mathematical convenience. $\mathbf{I}_m$ is an $m \times m$ identity matrix where m is the number of conductors.

$\mathbf{X}(z, s)$ can be expressed in a Taylor's series expansion around some point, s_0 , in the Laplace-domain:

$$\mathbf{X}(z, s) = \mathbf{W}^{(0)}(z, s_0) + \mathbf{W}^{(1)}(z, s_0)(s - s_0) + \mathbf{W}^{(2)}(z, s_0)(s - s_0)^2 + \dots \quad (4.44)$$

where $\mathbf{W}^{(i)}(z, s_0)$ is the i^{th} moment of $\mathbf{X}(z, s)$ computed at $s = s_0$, i.e.,

$$\mathbf{W}^{(i)}(z, s_0) = \frac{1}{i!} \frac{\partial^i}{\partial s^i} \mathbf{X}(z, s) \Big|_{s=s_0} \quad (4.45)$$

It can be noted from Equations (4.45) that the moments are functions of the spatial variable z . This is in contrast to the case of networks described by ordinary differential equations, where the moments are simply elements of the classical Euclidian space. The space of the moments in (4.44) can be rather described as a generalized Hilbert space [93] and may be denoted by $\mathcal{L}^{2m}(0, d)$.

Similar to Krylov-subspace techniques, the method based on Integrated Congruence Transform (ICT) proceeds by constructing an orthonormal basis for the moments. For that purpose, a classical orthogonalization procedure such as Modified Gram-Schmidt (MGS) is adequate. However, MGS requires a proper definition for the inner-product and norm mappings. The mappings that have been used for this purpose are defined in the following equations [1]:

$$\langle \mathbf{u}(z), \mathbf{v}(z) \rangle = \int_0^d \mathbf{u}^T(z) \mathbf{v}(z) dz \quad (4.46)$$

for the inner-product of any two elements $\mathbf{u}(z), \mathbf{v}(z) \in \mathcal{L}^{2m}(0, d)$, and

$$\|\mathbf{u}(z)\| = \int_0^1 \mathbf{u}^T(z) \mathbf{u}(z) dz \quad (4.47)$$

for the norm of any element $\mathbf{u}(z) \in \mathcal{L}^{2m}(0, d)$. To compute an orthonormal basis, $\mathbf{Q}(z)$, for the moments $\mathbf{W}^{(i)}(z, s_0)$, where $i = 0, \dots, q$, the algorithm first computes these moments explicitly. Computing the moments is accomplished by substituting from Equation (4.44) into (4.42) and equating similar powers of s to obtain the recursive relation

$$\left[T \frac{d}{dz} + N(z) + s_0 M(z) \right] \mathbf{W}^{(0)}(z, s_0) = \mathbf{0} \quad (4.48)$$

$$\left[T \frac{d}{dz} + N(z) + s_0 M(z) \right] \mathbf{W}^{(i)}(z, s_0) = -M(z) \mathbf{W}^{(i-1)}(z, s_0) \quad (4.49)$$

Once the moments are computed from (4.48) and (4.49) iteratively, they are passed to the MGS algorithm to construct an orthonormal basis $\mathbf{Q}(z)$. Figure 4.6 displays a pseudo-code procedure for the MGS process.

• Step 2: System Reduction

The system of partial differential equations of (4.42) is reduced using the integrated congruence transform.

$$(\hat{T} + \hat{N}_1 + \hat{M}) \hat{x}(s) = 0 \quad (4.50)$$

Algorithm 4.3.2: MGS METHOD($W^{(0)}(z), i = 0, \dots, q$)

```

 $q^{(0)}(z) \leftarrow \|W^{(i)}(z)\|$ 
for  $j \leftarrow 1$  to  $q$ 
     $\lambda(z) \leftarrow W^{(j)}(z)$ 
    for  $i \leftarrow 0$  to  $j - 1$ 
        do  $\left\{ \begin{array}{l} h \leftarrow \langle \lambda(z), q^{(i)}(z) \rangle \\ \lambda(z) \leftarrow \lambda(z) - q^{(i)}(z)h \end{array} \right.$ 
     $q^{(j)}(z) \leftarrow \lambda(z)$ 
 $Q(z) \leftarrow [q^{(0)}(z) \quad q^{(1)}(z) \quad \dots \quad q^{(q)}(z)]$ 
return ( $Q(z)$ )

```

Figure 4.6: Pseudo-code description of MGS process.

where

$$\hat{N}_1 = d \int_0^1 Q^T(z) N(z) Q(z) dz \quad (4.51)$$

$$\hat{M} = d \int_0^1 Q^T(z) M(z) Q(z) dz \quad (4.52)$$

$$\hat{T} = \int_0^1 Q^T T \frac{d}{dz} Q(z) dz \quad (4.53)$$

Note that in (4.50) the derivative w.r.t. the spatial variable has disappeared, reducing the Telegrapher's equations into a set of algebraic equations in the Laplace-domain. It has been shown in [1] that the system in (4.50) represents a system whose Y -parameters are given by

$$\hat{Y}(s) = \hat{b}^T (s\hat{M} + \hat{N})^{-1} \hat{b} \quad (4.54)$$

where

$$\hat{N} = \hat{N}_1 + \hat{N}_2, \quad (4.55)$$

$$\hat{N}_2 = \hat{T} - P, \quad (4.56)$$

$$P = Q_i^T(1)Q_v(1) - Q_i^T(0)Q_v(0) \quad (4.57)$$

$$\hat{b} = \begin{bmatrix} Q_i(0) \\ -Q_i(1) \end{bmatrix}^T \quad (4.58)$$

with $Q_i(z)$ and $Q_v(z)$ are the portions of the basis Q corresponding to the currents and voltages, respectively.

4.4 Discussion

This section has presented a general overview of the basic techniques to simulate high-speed interconnects in the time-domain in the presence of nonlinear terminations. Recently, the issue of passivity has been the main focus of these techniques. It was shown that MOR techniques based on Krylov subspace can be made to produce passive networks through using the Congruent transform.

It has been mentioned that Krylov subspace MOR techniques offers an additional numerical advantage. That advantage results from the fact that moments of the impulse response do not need not to be computed explicitly in order to obtain the desired approximation. Using higher order moments based on single point expansion, therefore, adds further accuracy and matches a wider frequency range.

A test example has been considered to further illustrate this point. In this test, a transmission line network of 3 conductors has been approximated using a suitable number of RLC lumped sections. However, instead of using the Arnoldi process to compute the orthogonal basis Q used in the reduction process, Q was constructed by computing the first moments *explicitly*

and then using the Householder algorithm to generate a spanning orthogonal basis for them. Figure 4.7 shows the performance of the PRIMA algorithm under these conditions for reduced systems of sizes 20, 30, 300, respectively. As can be seen from Figure 4.7, using high-order moments could not help in enhancing the accuracy of the reduced system. On the other hand, Figure 4.8 shows that a reduced system of size 35 was capable of capturing the whole frequency response when the PRIMA algorithm was implemented through using the Arnoldi process to construct the orthogonal basis Q .

Figure 4.7: MOR using PRIMA algorithm with *explicit* moment matching.

MOR based on ICT has been shown to produce guaranteed passive reduced-order models [1]. However, having to compute the moments explicitly before the orthogonalization step represents a numerical disadvantage. This is actually due to the fact that the moments are

Figure 4.8: MOR using PRIMA algorithm with *implicit* moment matching.

related through a *differential* operator as shown in (4.47) and (4.48). This problem will also be addressed in Chapter 7 in the context of linear time-varying systems.

Chapter 5

Passive Model Reduction of Distributed Interconnects

In the PRIMA algorithm explained in Section 4.3, it was demonstrated that obtaining passive reduced-order systems can be accomplished by applying congruent transform on both the G and C matrices provided that they contain only stamps coming out of lumped RLC components and independent voltage and current sources. In addition, both G and C matrices have to conform to a specific MNA formulation which was illustrated in Chapter 2.

This chapter proposes a new algorithm for passive model-order reduction of distributed interconnects. The proposed algorithm uses congruent transform on the G and C matrices to obtain the reduced-order system. The contribution of the proposed algorithm is that, in addition to stamps of lumped RLC components, the G and C matrices can contain stamps specific to lossy MTL without compromising the guarantee of the passivity property. The stamps developed for MTL are based on the Matrix-Pad macromodelling technique explained in Section 4.2.3. This approach offers the advantage of using the closed-form feature of the Matrix-Pad approach in representing any general uniform MTL independently of the of the PUL parameter matrices. A formal proof of the passivity of the reduced-order system is also

presented in this chapter.

5.1 Development of MNA Stamp

Section 4.2.3 demonstrated that an MTL can be represented by a set of cascaded subsections whose terminal voltages and currents are related by (4.15)–(4.18). Finding an MNA representation for the entire MTL requires finding a stamp for each subsection individually. It was shown in [94] that the Y -parameters for each subsection in (4.16) and (4.17) are given by:

$$\begin{aligned}
 Y_{11} = Y_{22} &= \frac{x_i^2 + y_i^2}{4x_id}(\mathbf{R} + s\mathbf{L})^{-1} + \frac{(\mathbf{G} + s\mathbf{C})d}{4x_i} \\
 &+ x_i \left((\mathbf{R} + s\mathbf{L})d + \frac{(x_i^2 + y_i^2)}{d}(\mathbf{G} + s\mathbf{C})^{-1} \right)^{-1} \\
 Y_{12} = Y_{21} &= \frac{x_i^2 + y_i^2}{4x_id}(\mathbf{R} + s\mathbf{L})^{-1} + \frac{(\mathbf{G} + s\mathbf{C})d}{4x_i} \\
 &- x_i \left((\mathbf{R} + s\mathbf{L})d + \frac{(x_i^2 + y_i^2)}{d}(\mathbf{G} + s\mathbf{C})^{-1} \right)^{-1}
 \end{aligned} \tag{5.1}$$

if the subsection corresponds to a complex pole-zero pair, and by:

$$\begin{aligned}
 Y_{11} &= Y_{22} = \frac{1}{2d}a_0(\mathbf{R} + s\mathbf{L})^{-1} + \frac{d}{2a_0}(\mathbf{G} + s\mathbf{C}) \\
 Y_{12} &= Y_{21} = -\frac{d}{2}a_0(\mathbf{R} + s\mathbf{L})^{-1} + \frac{d}{2a_0}(\mathbf{G} + s\mathbf{C})
 \end{aligned} \tag{5.2}$$

if the subsection corresponds to a real pole-zero pair.

Obtaining an MNA stamp for each subsection is done by realizing the Y -parameters in each subsection with a network. This realization has been performed in [95], [80]. Figures 5.1 and 5.2 present the realization networks for a single-conductor transmission line.

Figure 5.1: Realization of the Y -parameters in (5.1) of a 1-conductor transmission line. Figure 5.2: Realization of the Y -parameters in (5.2) of a 1-conductor transmission line.

Based on the networks shown in Figures 5.1 and 5.2, an MNA stamp can be developed for the MTL. The stamp will be composed of two components, namely, G_i and C_i , that are, respectively, plugged in the G and C matrices of the rest of the linear network. A subsection's stamps G_i and C_i are, respectively, given by

$$G_i = \begin{bmatrix} \left(\frac{x_i}{d} + \frac{\rho_i^2}{4x_i d}\right) R^{-1} + \frac{d}{4x_i} G & \frac{-x_i}{d} R^{-1} & 0 & \frac{d}{4x_i} G & \frac{\rho_i^2}{4x_i d} R^{-1} & 0 & 0 \\ \frac{-x_i}{d} R^{-1} & \frac{x_i}{d} R^{-1} & 0 & 0 & 0 & U & 0 \\ 0 & 0 & \frac{x_i d}{\rho_i^2} G & \frac{-x_i d}{\rho_i^2} G & 0 & -U & 0 \\ \frac{d}{4x_i} G & 0 & \frac{-x_i d}{\rho_i^2} G & \left(\frac{x_i d}{\rho_i^2} + \frac{d}{4x_i}\right) G & 0 & 0 & U \\ \frac{\rho_i^2}{4x_i d} R^{-1} & 0 & 0 & 0 & \frac{\rho_i^2}{4x_i d} R^{-1} & 0 & -U \\ 0 & -U & U & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -U & U & 0 & 0 \end{bmatrix} \quad (5.3)$$

$$C_i = \begin{bmatrix} \frac{d}{4x_i} C & 0 & 0 & \frac{d}{4x_i} C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{x_i d}{\rho_i^2} C & \frac{-x_i d}{\rho_i^2} C & 0 & 0 & 0 \\ \frac{d}{4x_i} C & 0 & \frac{-x_i d}{\rho_i^2} C & \left(\frac{x_i d}{\rho_i^2} + \frac{d}{4x_i}\right) C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{d}{x_i} L & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{4x_i d}{\rho_i^2} L \end{bmatrix} \quad (5.4)$$

for the subsections described by the Y -parameters of (5.1), and

$$G_i = \begin{bmatrix} \frac{d}{2a_0}G & 0 & \frac{d}{2a_0}G & U \\ 0 & \frac{a_0}{2d}R^{-1} & \frac{-a_0}{2d}R^{-1} & -U \\ \frac{d}{2a_0}G & \frac{-a_0}{2d}R^{-1} & \left(\frac{a_0}{2d}R^{-1} + \frac{d}{2a_0}G\right) & 0 \\ U & -U & 0 & 0 \end{bmatrix} \quad (5.5)$$

$$C_i = \begin{bmatrix} \frac{d}{2a_0}C & 0 & \frac{d}{2a_0}C & 0 \\ 0 & 0 & 0 & 0 \\ \frac{d}{2a_0}C & 0 & \frac{d}{2a_0}C & 0 \\ 0 & 0 & 0 & \frac{2d}{a_0}L \end{bmatrix} \quad (5.6)$$

for the subsection described by the Y -parameters of (5.2). Note that the size of the matrices G_i and C_i is $7m \times 7m$ if they are given by Equations (5.3) and (5.4) and is $4m \times 4m$ if they are given by Equations (5.5) and (5.6).

The concept of using a selector matrix to map the stamp of an element into the rest of the nodes space of the network will be illustrated through a simple example of a resistor stamp. As was mentioned in Chapter 2, the stamp of a resistor is obtained by adding to the G matrix 4 elements at the indices i, j corresponding to the nodes of the resistor as shown in Figure 5.3.

The process of adding a stamp for a resistor with conductance $g = \frac{1}{R}$ can be formalized as follows

$$G \leftarrow G + \Xi^T \begin{bmatrix} 1 \\ R \end{bmatrix} \Xi \quad (5.7)$$

In (5.7), Ξ is an $N \times 1$ vector defined as

$$\Xi = \begin{bmatrix} 0 & \cdots & 0 & \underbrace{1}_{i\text{-th position}} & 0 & \cdots & 0 & \underbrace{-1}_{j\text{-th position}} & 0 & \cdots & 0 \end{bmatrix}^T \quad (5.8)$$

Similarly, denote the selector matrix that maps the i -th subsection's stamp to the rest of the network space by $\Xi_i = [\xi_{m,n}]$, where $\xi_{m,n} \in \{-1, 0, 1\}$, $m \in \{1, \dots, N_\phi\}$, $n \in \{1, \dots, 7m\}$ if the subsection is corresponding to a complex pole-zero pair, (5.1), or $n \in \{1, \dots, 4m\}$ if the

$$\begin{array}{c}
 \begin{array}{cc}
 & i & & j \\
 & \vdots & & \vdots \\
 & 0 & & 0 \\
 i & \cdots & 0 & g & 0 & \cdots & 0 & -g & 0 & \cdots \\
 & & & 0 & & & & 0 & & \\
 & & & \vdots & & \cdots & & \vdots & & \\
 & & & 0 & & & & 0 & & \\
 j & \cdots & 0 & -g & 0 & \cdots & 0 & g & 0 & \cdots \\
 & & & 0 & & & & 0 & & \\
 & & & \vdots & & & & \vdots & &
 \end{array}
 \end{array}
 \left[\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right]$$

Figure 5.3: A Resistor's stamp.

subsection is corresponding to a real pole-zero, (5.2), and N_ϕ is the total number of variables in the network including the extra state-variables augmented by the stamps of each subsection.

Based on the above development, the generalized MNA formulation for a network with N_t distributed interconnect sets can be described as follows,

$$(M + sN)X_a(s) = B_a v_p, \quad i_p = B_a^T X_a(s) \quad (5.9)$$

where

$$M = \begin{bmatrix} G_\phi & 0 \\ 0 & 0 \end{bmatrix} + \sum_{k=1}^{N_t} \sum_i (\Xi_i^k)^T G_i^k \Xi_i^k \quad (5.10)$$

$$N = \begin{bmatrix} C_\phi & 0 \\ 0 & 0 \end{bmatrix} + \sum_{k=1}^{N_t} \sum_i (\Xi_i^k)^T C_i^k \Xi_i^k \quad (5.11)$$

and G_ϕ and C_ϕ are matrices that contain stamps of lumped RLC components. The indices i and k represent the i^{th} subsection of the k^{th} interconnect. In the next section, we describe the model reduction algorithm for reducing the system described by (5.9).

5.2 Model Reduction and Passivity Preservation

This section describes a reduction algorithm for distributed interconnects based on the congruent transformation. In addition, a rigorous proof of passivity of the reduced-order macromodel is presented.

5.2.1 Model Reduction

An orthonormal matrix Q is constructed using $\lfloor q/p \rfloor$ iterations of the block Arnoldi algorithm [91], as shown in Figure 4.5, such that

$$\begin{aligned} \text{colsp}(Q) &= Kr(A, R, q) \\ Q^T Q &= I_q \end{aligned} \quad (5.12)$$

where

$$A = -M^{-1}N, \quad R = M^{-1}B \quad (5.13)$$

$$Kr(A, R, q) = \text{colsp}[R, AR, A^2R, \dots, A^{k-1}R]; \quad (k = \lfloor q/p \rfloor) \quad (5.14)$$

and p is the number of ports and q is the number of iterations used to run the Arnoldi algorithm. Next, the matrix Q is used to reduce the augmented system matrices of (5.9) by the congruence transform

$$\hat{M} = Q^T M Q, \quad \hat{N} = Q^T N Q, \quad \hat{B}_a = Q^T B_a \quad (5.15)$$

The admittance matrix for the reduced system is given by

$$\hat{Y}(s) = \hat{B}_a^T (\hat{M} + s\hat{N})^{-1} \hat{B}_a \quad (5.16)$$

It can be shown that the reduced-order system described by (5.15) and (5.16) preserves the first $\lfloor q/p \rfloor$ block moments. The proof of the preservation of moments is identical to the one given

in [15], where the matrices M and N contain stamps of lumped components only. However, a new approach is needed here to prove that the reduced system is passive since the matrices M and N of the original system contain the stamps (5.3)-(5.6) of transmission lines.

5.2.2 Passivity Preservation

In this section we prove that the reduced-model represented by (5.16) is passive. The sufficient and necessary conditions required for the reduced system of (5.16) to be passive are

1. $\hat{Y}(s^*) = \hat{Y}^*(s)$, and
2. $\hat{Y}$ is a positive real matrix, that is $z^{*T}[\hat{Y}^T(s^*) + \hat{Y}(s)]z \geq 0$ for all complex values of s satisfying that $\Re(s) > 0$ and any arbitrary complex vector z .

The first condition is automatically satisfied since the reduced matrices $\hat{M}$, $\hat{N}$ and $\hat{B}_a$ are all real. To show that second condition is also satisfied we set $\hat{Y}_h(s) = \hat{Y}(s) + \hat{Y}^T(s^*)$ and then handle the quadratic form $z^{*T}\hat{Y}_h(s)z$ by some vector algebra to obtain

$$\begin{aligned} z^{*T}\hat{Y}_h(s)z &= z^{*T}\hat{B}_a^T[(\hat{M} + s\hat{N})^{-1} + (\hat{M} + s^*\hat{N})^{-T}]\hat{B}_a z \\ &= z^{*T}\hat{B}_a^T(\hat{M} + s\hat{N})^{-1}[(\hat{M} + s\hat{N}) + (\hat{M} + s^*\hat{N})^T](\hat{M} + s^*\hat{N})^{-T}\hat{B}_a z \end{aligned} \quad (5.17)$$

Setting $\Phi = (\hat{M} + s^*\hat{N})^{-T}\hat{B}_a z$ and $s = \sigma + j\omega$ yields

$$\begin{aligned} z^{*T}\hat{Y}_h(s)z &= \Phi^{*T}[\hat{M} + \hat{M}^T + \sigma(\hat{N} + \hat{N}^T)]\Phi \\ &= \Phi^{*T}Q^T[M + M^T + \sigma(N + N^T)]Q\Phi \\ &= y^{*T}[M + M^T]y + y^{*T}[\sigma(N + N^T)]y \end{aligned} \quad (5.18)$$

where $\mathbf{y} = \mathbf{Q}\Phi$. Substituting from (5.10) and (5.11) for the matrices $\mathbf{M}$ and $\mathbf{N}$ in (5.18) and using the fact that $\mathbf{C}_\phi$ is symmetric yields

$$\begin{aligned} \mathbf{z}^{*T} \hat{\mathbf{Y}}_h(s) \mathbf{z} &= \mathbf{y}^{*T} \begin{bmatrix} \mathbf{G}_\phi + \mathbf{G}_\phi^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{y} + 2\mathbf{y}^{*T} \sigma \begin{bmatrix} \mathbf{C}_\phi & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{y} \\ &+ \mathbf{y}^{*T} \left(\sum_{k=1}^{N_t} \sum_i (\Xi_i^k)^T (\mathbf{G}_i^k + (\mathbf{G}_i^k)^T) \Xi_i^k \right) \mathbf{y} \\ &+ \mathbf{y}^{*T} \sigma \left(\sum_{k=1}^{N_t} \sum_i (\Xi_i^k)^T (\mathbf{C}_i^k + (\mathbf{C}_i^k)^T) \Xi_i^k \right) \mathbf{y} \end{aligned} \quad (5.19)$$

Thus, to prove that the second condition is satisfied we need to show that each one of the four quadratic forms in (5.17) is nonnegative. Firstly, we consider the first two quadratic forms. The matrices $\mathbf{C}_\phi$ and $\mathbf{G}_\phi + \mathbf{G}_\phi^T$ can be formulated to be nonnegative definite [15]. Hence,

$$\begin{aligned} \mathbf{y}^{*T} \begin{bmatrix} \mathbf{G}_\phi + \mathbf{G}_\phi^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{y} &\geq 0 \\ 2\mathbf{y}^{*T} \sigma \begin{bmatrix} \mathbf{C}_\phi & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{y} &\geq 0 \end{aligned} \quad (5.20)$$

On the other hand, proving that the last two quadratic forms in (5.19) are nonnegative requires showing that the matrices $(\Xi_i^k)^T (\mathbf{G}_i^k + (\mathbf{G}_i^k)^T) \Xi_i^k$ and $(\Xi_i^k)^T (\mathbf{C}_i^k + (\mathbf{C}_i^k)^T) \Xi_i^k$ are nonnegative definite. However, these matrices are described in the form of a congruence transform of the matrices $\mathbf{G}_i^k + (\mathbf{G}_i^k)^T$ and $\mathbf{C}_i^k + (\mathbf{C}_i^k)^T$, respectively, where Ξ_i^T is used as the transformation matrix. Hence, $(\Xi_i^k)^T (\mathbf{G}_i^k + (\mathbf{G}_i^k)^T) \Xi_i^k$ and $(\Xi_i^k)^T (\mathbf{C}_i^k + (\mathbf{C}_i^k)^T) \Xi_i^k$ are nonnegative definite if $\mathbf{G}_i^k + (\mathbf{G}_i^k)^T$ and $\mathbf{C}_i^k + (\mathbf{C}_i^k)^T$ are nonnegative definite [96].

To complete the proof of passivity we need to show that $\mathbf{G}_i^k + (\mathbf{G}_i^k)^T$ and $\mathbf{C}_i^k + (\mathbf{C}_i^k)^T$ are nonnegative definite. This proof is given in the next subsection by establishing the basic properties of the matrices $\mathbf{G}_i^k$ and $\mathbf{C}_i^k$.

Properties of G_i and C_i Matrices

For ease of manipulation the superscript k in G_i^k and C_i^k has been omitted. We only consider the case where these matrices represent a complex pole-zero subsection as in (5.3) and (5.4) since similar arguments can be used to treat the real pole-zero subsection of (5.5) and (5.6). G_i and C_i can be represented as,

$$G_i = \begin{bmatrix} W_i & E \\ -E^T & 0 \end{bmatrix}, \quad C_i = \begin{bmatrix} Z_i & 0 \\ 0 & H_i \end{bmatrix} \quad (5.21)$$

where

$$\begin{aligned} W_i &= \begin{bmatrix} \left(\frac{x_i}{d} + \frac{\rho_i^2}{4x_i d} \right) R^{-1} + \frac{d}{4x_i} G & \frac{-x_i}{d} R^{-1} & 0 & \frac{d}{4x_i} G & \frac{\rho_i^2}{4x_i d} R^{-1} \\ \frac{-x_i}{d} R^{-1} & \frac{x_i}{d} R^{-1} & 0 & 0 & 0 \\ 0 & 0 & \frac{x_i d}{\rho_i^2} G & \frac{-x_i d}{\rho_i^2} G & 0 \\ \frac{d}{4x_i} G & 0 & \frac{-x_i d}{\rho_i^2} G & \left(\frac{x_i d}{\rho_i^2} + \frac{d}{4x_i} \right) G & 0 \\ \frac{\rho_i^2}{4x_i d} R^{-1} & 0 & 0 & 0 & \frac{\rho_i^2}{4x_i d} R^{-1} \end{bmatrix} \\ Z_i &= \begin{bmatrix} \frac{d}{4x_i} C & 0 & 0 & \frac{d}{4x_i} C & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{x_i d}{\rho_i^2} C & \frac{-x_i d}{\rho_i^2} C & 0 \\ \frac{d}{4x_i} C & 0 & \frac{-x_i d}{\rho_i^2} C & \left(\frac{x_i d}{\rho_i^2} + \frac{d}{4x_i} \right) C & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ E^T &= \begin{bmatrix} 0 & U & -U & 0 & 0 \\ 0 & 0 & 0 & U & -U \end{bmatrix} \quad H_i = \begin{bmatrix} \frac{d}{x_i} L & 0 \\ 0 & \frac{4x_i d}{\rho_i^2} L \end{bmatrix} \end{aligned} \quad (5.22)$$

Clearly H_i is a symmetric nonnegative definite matrix since it is block diagonal with symmetric nonnegative definite diagonal blocks [96]. The next two theorems are developed to show that W_i and Z_i are nonnegative definite.

Theorem 5.2.1 *Let Θ be a block structured matrix that has only 4 nonzero block matrices located at the block entries (i, i) , (i, j) , (j, i) and (j, j) . Assume that these four blocks are equal to K , i.e.,*

$$\Theta = \begin{bmatrix} 0 & \dots & 0 \\ & K & \dots & K \\ & \vdots & & \vdots \\ & K & \dots & K \\ 0 & \dots & 0 \end{bmatrix} \quad (5.23)$$

where $K \in \mathbb{R}^{q \times q}$ is a nonnegative definite matrix. Then Θ is nonnegative definite.

Proof To simplify the proof considerably, consider a permuted version of Θ , Θ_P , where

$$\Theta = P^T \Theta_P P \quad (5.24)$$

and P is some suitable permutation matrix that is chosen such that,

$$\Theta_P = \begin{bmatrix} K & K & 0 & \dots & 0 \\ K & K & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (5.25)$$

Θ_P can be expressed in terms of congruence transform as

$$\Theta_P = F^T \text{diag}(K, 0, \dots, 0) F \quad (5.26)$$

where

$$F = \begin{bmatrix} U & U & 0 & \dots & 0 \\ 0 & U & 0 & \dots & 0 \\ 0 & 0 & U & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & U \end{bmatrix} \quad (5.27)$$

From (5.24)-(5.27), the matrix Θ can be written in the form of the congruence transform

$$\Theta = (FP)^T \text{diag}(K, 0, \dots, 0)(FP) \quad (5.28)$$

Since K is nonnegative definite, then $\text{diag}(K, 0, \dots, 0)$ is also nonnegative definite. Hence, the matrix Θ is nonnegative definite since it is defined as a congruence transform of $\text{diag}(K, 0, \dots, 0)$ using FP as the transformation operator [96]. ■

Theorem 5.2.2 *Let Ψ be a matrix that has the same structure as Θ except that the blocks at the entries (i, j) and (j, i) are negated,*

$$\Psi = \begin{bmatrix} 0 & \dots & 0 \\ & K & \dots & -K \\ & \vdots & & \vdots \\ & -K & \dots & K \\ 0 & \dots & 0 \end{bmatrix} \quad (5.29)$$

where $K \in \mathbb{R}^{q \times q}$ is a nonnegative definite matrix. Then Ψ is nonnegative definite.

Proof The proof of this theorem follows similar steps to the proof of the above theorem where in this case we consider the permuted matrix Ψ_P , to put Ψ in the form of the congruence transform,

$$\Theta = P^T \Psi_P P \quad (5.30)$$

where P is some suitable permutation matrix such that

$$\Psi_P = \begin{bmatrix} K & -K & 0 & \dots & 0 \\ -K & K & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (5.31)$$

Ψ_P can be expressed in the form

$$\Psi_P = F^T \text{diag}(K, 0, \dots, 0) F \quad (5.32)$$

where

$$F = \begin{bmatrix} U & -U & 0 & \dots & 0 \\ 0 & U & 0 & \dots & 0 \\ 0 & 0 & U & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & U \end{bmatrix} \quad (5.33)$$

From (5.30)-(5.33), Ψ can be written in terms of congruence transform

$$\Psi = (FP)^T \text{diag}(K, 0, \dots, 0) (FP) \quad (5.34)$$

Since K is nonnegative definite, the matrix $\text{diag}(K, 0, \dots, 0)$ is also nonnegative definite.

Hence, Ψ is nonnegative definite since it is defined as congruence transform of $\text{diag}(K, 0, \dots, 0)$ using FP as the transformation operator. ■

Corollary 5.2.3 *The matrices W_i and Z_i in (5.22) are symmetric nonnegative definite.*

Proof : W_i and Z_i can be written as

$$W_i = W_{i,1} + W_{i,2} + W_{i,3} + W_{i,4} \quad Z_i = Z_{i,1} + Z_{i,2} \quad (5.35)$$

where

$$W_{i,1} = \begin{bmatrix} \frac{\rho_i^2}{4x_id} R^{-1} & 0 & 0 & 0 & \frac{\rho_i^2}{4x_id} R^{-1} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{\rho_i^2}{4x_id} R^{-1} & 0 & 0 & 0 & \frac{\rho_i^2}{4x_id} R^{-1} \end{bmatrix} \quad (5.36)$$

$$W_{i,2} = \begin{bmatrix} \frac{x_i}{d} R^{-1} & -\frac{x_i}{d} R^{-1} & 0 & 0 & 0 \\ -\frac{x_i}{d} R^{-1} & \frac{x_i}{d} R^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.37)$$

$$W_{i,3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{x_i d}{\rho_i^2} G & -\frac{x_i d}{\rho_i^2} G & 0 \\ 0 & 0 & -\frac{x_i d}{\rho_i^2} G & \frac{x_i d}{\rho_i^2} G & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.38)$$

$$W_{i,4} = \begin{bmatrix} \frac{d}{4x_i} G & 0 & 0 & \frac{d}{4x_i} G & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{d}{4x_i} G & 0 & 0 & \frac{d}{4x_i} G & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.39)$$

$$Z_{i,1} = \begin{bmatrix} \frac{d}{4x_i} C & 0 & 0 & \frac{d}{4x_i} C & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{d}{4x_i} C & 0 & 0 & \frac{d}{4x_i} C & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.40)$$

$$Z_{i,2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{x_i d}{\rho_i^2} C & -\frac{x_i d}{\rho_i^2} C & 0 \\ 0 & 0 & -\frac{x_i d}{\rho_i^2} C & \frac{x_i d}{\rho_i^2} C & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.41)$$

Since the constants ρ_i^2 and x_i are positive and the PUL parameter matrices $\mathbf{R}$, $\mathbf{L}$, $\mathbf{G}$, and $\mathbf{C}$ are nonnegative definite, then using Theorems (5.2.1) and (5.2.2), all the block matrices in (5.36)-(5.41) are nonnegative definite. This means that $\mathbf{W}_i$ and $\mathbf{Z}_i$ are the summation of symmetric nonnegative definite matrices, and hence they are symmetric nonnegative definite [96]. ■

Corollary 5.2.4 *The matrices $(\mathbf{C}_i^k + \mathbf{C}_i^k)^T$ and $(\mathbf{G}_i^k + \mathbf{G}_i^k)^T$ are symmetric nonnegative definite.*

Proof This result can be easily deduced by noting that

$$(\mathbf{C}_i^k + \mathbf{C}_i^k)^T = \begin{bmatrix} \mathbf{Z}_i & 0 \\ 0 & \mathbf{H}_i \end{bmatrix} + \begin{bmatrix} \mathbf{Z}_i & 0 \\ 0 & \mathbf{H}_i \end{bmatrix}^T = \begin{bmatrix} 2\mathbf{Z}_i & 0 \\ 0 & 2\mathbf{H}_i \end{bmatrix} \quad (5.42)$$

$$(\mathbf{G}_i^k + \mathbf{G}_i^k)^T = \begin{bmatrix} \mathbf{W}_i & \mathbf{E} \\ -\mathbf{E}^T & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{W}_i & \mathbf{E} \\ -\mathbf{E}^T & 0 \end{bmatrix}^T = \begin{bmatrix} 2\mathbf{W}_i & 0 \\ 0 & 0 \end{bmatrix} \quad (5.43)$$

which means that $(\mathbf{C}_i^k + \mathbf{C}_i^k)^T$ and $(\mathbf{G}_i^k + \mathbf{G}_i^k)^T$ are block diagonals with symmetric nonnegative definite diagonal blocks. This concludes the proof of passivity. ■

5.3 Computational Results

Two examples are presented in this section to demonstrate the validity and efficiency of the proposed method. The second example includes interconnects with frequency dependent param-

eters. Handling transmission lines with frequency-dependent parameters has been addressed in [80]. The results given by the proposed method are compared with SPICE [17] analysis.

5.3.1 Example 1

A two port linear subnetwork consisting of 1516 linear components (including 30 transmission lines) with nonlinear terminations has been considered for this example. A stamp representing a rational approximation of order (5/5) was used for the transmission lines. The original set of MNA equations contained a total of 1682 variables. Using a multipoint version of the reduction algorithm of Section 5.2.1 [23], the size of the reduced system obtained was 66x66. Figures 5.5 and 5.6 show a comparison of Y -parameters of the linear subnetwork, that are obtained using SPICE with conventional lumped RLC segmentation approach [3] as well as the proposed macromodel. Figures 5.7 and 5.8 present a comparison of time responses, at two output nodes of the circuit (V_{p2} and V_{out}), respectively. The input pulse used for this example has a rise/fall time of 0.02 ns and pulse width of 5 ns. The transient simulation of the reduced-order system on a Sun Ultra 20 machine required 30 seconds of CPU time while the conventional lumped system required 817 seconds on the same machine (Table 5.1).

proposed Method (sec.)	Conventional Lumped Model (sec.)	Speed-up
30	817	27

Table 5.1: CPU comparison of time-domain response for example 1.

Figure 5.4: A linear subnetwork circuit with nonlinear termination.

5.3.2 Example 2

A two port linear subnetwork consisting of 2203 linear components (including 35 interconnects with frequency-dependent parameters) with nonlinear terminations is considered. Figure shows the PUL resistance and inductance as function of frequency. A Padé approximation (order 8/8) is used to model each interconnect. The total size of the MNA equations including the Padé interconnect stamps is 4352 variables. Performing the passive reduction scheme, a macromodel containing 184 variables is obtained. Figures 5.11- 5.12 show a comparison of Y-parameters of the linear subnetwork, that are obtained using SPICE with conventional lumped RLC segmentation approach [3] as well as the proposed macromodel. Figures 5.14 and 5.15 present a comparison of time responses, at output nodes V_{p2} and V_{out} , respectively. The input pulse used for this example has a rise/fall time of 0.02ns and pulse width of 5ns. On a SunUltra-5 computer it takes 1.7 minutes to obtain the transient response with the reduced model, while the conventional lumped system requires 80.5 minutes (Table 5.2).

proposed Method (min.)	Conventional Lumped Model (min.)	Speed-up
1.7	80.5	47

Table 5.2: CPU comparison of time response for example 2.

Figure 5.5: Magnitude of Y_{11} of linear subnetwork for example 5.3.1

Figure 5.6: Magnitude of Y_{12} of linear subnetwork for example 5.3.1

Figure 5.7: Magnitude of Y_{22} of linear subnetwork for example 5.3.1 Figure 5.8: Time domain response at output node V_{p2} for example 5.3.1

Figure 5.10: R and L vs. frequency for example 5.3.2

5.4 Conclusion

A new algorithm is presented in this chapter to include transmission lines in passive model-reduction techniques. The transmission lines can be lossy, coupled and can include frequency-dependent parameters. The proposed scheme uses a closed-form Padé approximation to model

Figure 5.11: Magnitude of Y_{11} of linear subnetwork for example 5.3.2

each transmission line. In addition, the algorithm guarantees the passivity of the reduced system.

Figure 5.12: Magnitude of Y_{12} of linear subnetwork for example 5.3.2

Figure 5.13: Magnitude of Y_{22} of linear subnetwork for example 5.3.2

Figure 5.14: Time response at output port Vp2 for example 5.3.2

Figure 5.15: Time domain response at output node Vout for example 5.3.2

Chapter 6

Finding Steady-State Solution via Circuit Reduction

6.1 Introduction

The rapid growth in RF silicon ICs for mobile communication systems has placed new demands on the simulation tools. Quantities such as intermodulation and harmonic distortion are typically of interest to circuit designers. Determining these quantities is best achieved through steady-state analysis. However, finding the steady-state response for large analog and microwave nonlinear circuits represents an increasing challenge since these types of circuits exhibit characteristics that make most traditional transient-based time domain approaches perform poorly. For instance, if there are widely separated time constants in the circuit then the usual transient analysis can take many clock cycles before the steady-state solution is reached.

Chapter 3 reviewed the main approaches for finding the steady-state operating point in nonlinear circuits. In particular, the HB approach has been the focus of increasing interest over the past decade due to its ability to handle nonlinear elements in the frequency-domain. This makes including distributed elements quite natural since these elements are best described

in the frequency-domain as explained in Chapter 4. In addition, the HB approach can be extended easily to simulate circuits excited by non-commensurate multi-tone signals such as RF circuits [59].

The main philosophy behind HB is that waveforms in a periodically excited nonlinear circuit are periodic and thus can be represented in the frequency-domain as a finite Fourier series. This effectively transforms the differential equations describing the circuit into a set of nonlinear algebraic equations that can be solved directly using some iterative techniques such as the Newton Raphson (NR) method [49]. However, the main bottleneck that comes with solving this set of nonlinear equations is the prohibitive CPU cost that is required to store and factorize the Jacobian matrix of the nonlinear equations. Furthermore, for large RF integrated circuits with strong nonlinearities this Jacobian matrix becomes large and dense and its factorization can be expensive even for medium sized circuits [62]. In addition, like all other locally convergent methods, NR has convergence problems. Convergence can be achieved only by finding initial values sufficiently close to the solution, which is difficult particularly for the cases of the high excitation levels.

Continuation methods [27] on the other hand, have been proposed to avoid the convergence problems associated with the direct NR method. In the continuation schemes, the original set of nonlinear equations is replaced with an auxiliary system of nonlinear equations of the same size whose solution is trivial. Then using some sweeping parameter, the solution trajectory can be traced back to the original system. However, these methods are computationally expensive as they require the factorization of the Jacobian matrix corresponding to many values of the sweeping parameter.

Matrix-implicit linear solvers have been introduced to reduce the high cost of factorizing and the storing the Jacobian matrix. In this scheme, only the components of the Jacobian matrix need to be stored and Krylov-subspace iterative solvers are used [57]. However, these methods require the use of pre-conditioners to improve their convergence. In addition they still need to

operate on the full size of the Jacobian matrix.

In this chapter, a new algorithm that poses the problem from a different perspective is presented. The crux of the proposed algorithm is to reduce the size of the Jacobian matrix needed for the HB solution quite drastically, through the introduction of the new concept of nonlinear circuit reduction. This enables the original large set of nonlinear equations representing the HB equations to be replaced with a much smaller set of nonlinear equations. The main feature of the reduced set is that it shares with the original one the first few derivatives w.r.t. to the amplitude of the time-varying signal sources at the DC operating point. A continuation scheme is then used to trace the solution of the reduced equations and map it back to the space of the original circuit equations. This introduces a significant computational advantage as the size of the Jacobian matrix that needs to be factorized is typically much smaller than the size of the original Jacobian.

The organization of this chapter is as follows. Section 6.2 discusses the problem of solving a system of nonlinear equations and the idea of using continuation methods to overcome the convergence problems. Section 6.3 explains the proposed circuit reduction technique. Section 6.4 illustrates the computation of the subspace used in the circuit reduction. Section 6.5 discusses the computational complexity of the proposed algorithm. Section 6.6 presents a rigorous proof of the preservation of the derivatives in the reduced system. Numerical testing examples are presented in Section 6.7.

6.2 Solving Nonlinear equations

This section is devoted to studying the problem of finding the solution of a general set of nonlinear algebraic equations,

$$\Phi_h(\mathbf{X}) = 0 \tag{6.1}$$

where $\Phi_h(\mathbf{X})$ in the context of this chapter refers to the set of nonlinear equations resulting from applying the HB approaches. $\Phi_h(\mathbf{X})$ is reproduced here for completeness:

$$\Phi_h(\mathbf{X}) = \bar{\mathbf{G}}\mathbf{X} + j\Omega\bar{\mathbf{C}}\mathbf{X} + \mathbf{F}(\mathbf{X}) - \mathbf{B} \quad (6.2)$$

Section 3.2 describes the computation of the various terms in (6.2). Methods of finding a solution for equation (6.1) are explored in the following subsections.

6.2.1 Newton-Raphson (NR) Method

In the NR method the harmonics solution vector, $\mathbf{X}$, is searched via the iterative process,

$$\mathbf{X}^{(j+1)} = \mathbf{X}^{(j)} - [\mathbf{J}_h(\mathbf{X}^{(j)})]^{-1}\Phi_h(\mathbf{X}^{(j)}) \quad (6.3)$$

with $\mathbf{X}^{(0)}$ is an initial guess solution. Details of computing $\mathbf{J}_h(\mathbf{X}^{(j)})$ and $\Phi_h(\mathbf{X}^{(j)})$ have been explained in Section 3.2.5. As mentioned in section 3.2.2 the major time consuming part is spent in computing and factorizing the Jacobian matrix. In addition, it is widely known that the NR method is locally convergent. Thus, convergence is possible only if the starting point $\mathbf{X}^{(0)}$ is sufficiently close to the solution point.

6.2.2 Continuation Methods

Continuation methods (also known as *Homotopy* methods) have been introduced to overcome the convergence problems associated with the iterative techniques such as the NR method described above. The basic idea of the continuation methods is to augment the system $\Phi_h(\mathbf{X})$ in (6.1) with some parameter, say α , to obtain another auxiliary system $\Psi_h(\mathbf{X}, \alpha)$. Here α is chosen such that the system given by $\Psi_h(\mathbf{X}, 0)$ has a trivial solution in $\mathbf{X}$ and the system given at $\Psi_h(\mathbf{X}, 1) = \Phi_h(\mathbf{X})$ identically in $\mathbf{X}$. Continuation methods can be broadly classified into two categories based on the technique of embedding the continuation parameter, α , in $\Phi_h(\mathbf{X})$.

External Parameter Embedding

This approach of homotopy is based on using the parameter α externally to $\Phi_h(\mathbf{X})$. An example of this approach is obtained through formulating $\Psi_h(\mathbf{X})$ as

$$\Psi_h(\mathbf{X}) \equiv (1 - \alpha)(\mathbf{X} - \mathbf{X}_a) + \alpha\Phi_h(\mathbf{X}) \quad (6.4)$$

It is clear from (6.4) that at $\alpha = 0$, $\Psi_h(\mathbf{X})$ has a trivial solution given by

$$\mathbf{X} = \mathbf{X}_a \quad (6.5)$$

and at $\alpha = 1$

$$\Psi_h(\mathbf{X}) = \Phi_h(\mathbf{X}) \quad (6.6)$$

Note that in (6.4) the embedding parameter α is external to $\Phi_h(\mathbf{X})$.

Internal Parameter Embedding

This approach is also known as natural embedding. Here, the parameter α is embedded internally in $\Phi_h(\mathbf{X})$, whose solution is required. The choice of the parameter α in the natural embedding approach usually corresponds to some physical quantity in the system. Sweeping α in this case represents deforming the original system into a simple system with a trivial solution at $\alpha = 0$.

For the case of HB nonlinear equations, a natural choice for the parameter α would be the power of the time-varying forcing input. This can be achieved by splitting the harmonics source vector, $\mathbf{B}$, into two vectors and writing down the auxiliary system $\Psi_h(\mathbf{X}, \alpha)$ as

$$\Psi_h(\mathbf{X}, \alpha) = \bar{\mathbf{G}}\mathbf{X} + j\Omega\bar{\mathbf{C}}\mathbf{X} + \mathbf{F}(\mathbf{X}) + \mathbf{B}_{dc} - \alpha\mathbf{B}_{ac} \quad (6.7)$$

Here, $\mathbf{B}_{ac}$ is a vector representing the time-varying forcing input circuit and $\mathbf{B}_{dc}$ represents the biasing dc sources. At $\alpha = 0$, the solution of (6.7) is simply the solution obtained solution

from the usual DC analysis, while at $\alpha = 1$ the system $\Psi_h(\mathbf{X}, 1)$ has the same solution as $\Phi_h(\mathbf{X})$.

A typical continuation method proceeds by taking small increments in α and solving the system $\Psi_h(\mathbf{X}, \alpha)$ for many values of α until $\alpha = 1$.

Difficulties of Continuation Methods

Although continuation methods have provided a major advantage in being able to eliminate convergence problems, they are computationally expensive as they require the solution of many auxiliary systems of equations corresponding to numerous values of $\alpha \in (0, 1)$. This problem is even further aggravated for the case of HB equations since the HB Jacobian matrix is typically huge and coupled and its factorization overwhelms even medium sized computers.

6.3 Overview of the Circuit reduction Algorithm

This section describes a new algorithm that combines the advantage of using the continuation methods, and yet avoids the huge computational cost that comes as a result of the large size of equations with its coupled Jacobian matrix. The proposed approach is based on orthonormal projection methods through congruent transformation.

An overview of the proposed algorithm is shown in Figure 6.1. The first step of the algorithm consists of embedding a continuation parameter α in the HB equations $\Phi_h(\mathbf{X}) = 0$. This has been already formulated in (6.7). The system $\Psi_h(\mathbf{X}, \alpha)$ is then reduced to a much smaller system of equations $\hat{\Psi}_h(\hat{\mathbf{X}}, 1)$. Next, the solution of the reduced system is then traced as α is swept from 0 to 1 in the space of the *reduced system*. At $\alpha = 1$, the solution from the reduced space is then mapped back to the original system. The solution is therefore obtained entirely in the small space avoiding the high cost of solving the large system of nonlinear HB equations.

Figure 6.1: Schematic representation for the proposed algorithm.

6.3.1 Circuit Reduction

In this section, the second step of the reduction scheme shown in Figure 6.1 is presented. In this step, the large system of equations $\Psi(X_h, \alpha)$ is reduced into a much smaller system $\hat{\Psi}(\hat{X}, \alpha)$. The main objective of the reduction algorithm is to significantly reduce the size of the original circuit and, consequently, the size of the Jacobian matrix. The analysis is done in the reduced space, thus resulting in CPU cost savings. The reduced set of equations is obtained by effecting the following change of variables in the original set of Equation (6.7)

$$X \rightarrow Q_h \hat{X} \quad (6.8)$$

With the change of variables in (6.8) the augmented Harmonic Balance equations in (6.7) become

$$\bar{G}Q_h \hat{X} + j\Omega \bar{C}Q_h \hat{X} + F(Q_h \hat{X}) = B_{dc} + \alpha B_{ac} \quad (6.9)$$

Equation (6.9) is then pre-multiplied by Q_h^T , and the following reduced system is obtained

$$\hat{G}\hat{X} + \hat{C}\hat{X} + \hat{F}(\hat{X}) - \hat{B}_{dc} - \alpha\hat{B}_{ac} = 0 \quad (6.10)$$

where

$$\begin{aligned} \hat{G} &= Q_h^T \bar{G} Q_h \\ \hat{C} &= Q_h^T J \Omega \bar{C} Q_h \\ \hat{F}(\hat{X}) &= Q_h^T F(Q_h X) \\ \hat{B}_{dc} &= Q_h^T B_{dc} \end{aligned} \quad (6.11)$$

In the above equations, $Q_h \in \mathbb{R}^{N_{hb} \times N_q}$ is an orthonormal basis of the subspace spanned by the first N_q derivatives of X_h with respect to α , with $N_q \ll N_{hb}$. In other words, if the Taylor expansion of X_h with respect to α is

$$X = \sum_{k=0}^{\infty} A_k (\alpha - \alpha_0)^k \quad (6.12)$$

then

$$\text{colsp}[Q_h] = \text{colsp}[A_0, A_1, \dots, A_{q-1}] \quad (6.13)$$

The basic idea behind this change of variables is that the reduced system obtained by this transformation preserves the first N_q derivatives of X_h with respect to α under the mapping Q_h . More precisely, if $\hat{X}$ is expanded as the Taylor series

$$\hat{X} = \sum_{k=0}^{\infty} \hat{A}_k (\alpha - \alpha_0)^k \quad (6.14)$$

then

$$A_k = Q_h \hat{A}_k, \quad k = 0, \dots, q-1 \quad (6.15)$$

The proof of conservation of derivatives is shown in section 6.6. The solution of the reduced system is therefore expected to accurately approximate that of the original system as α is tracked from 0 to 1.

6.3.2 Solution of the Reduced Equations

After the augmented system of equations is reduced, the next step is to solve the reduced order system. As mentioned in the previous section, the CPU cost savings result from the fact that the analysis is done in the reduced system space, instead of solving the original large system. In this section the steps for solving the reduced system are presented. Consider the system of equations in (6.10)

$$\hat{\Psi}(\hat{X}, \alpha) = \hat{G}\hat{X} + \hat{C}\hat{X} + \hat{F}(\hat{X}) - \hat{B}_{dc} - \alpha\hat{B}_{ac}$$

In order to simplify the notation, the above system will be denoted as $\hat{\Psi}_i(\hat{X}) = \hat{\Psi}(\hat{X}, \alpha_i)$ when $\alpha = \alpha_i$. The solution of the reduced system of equations is found by applying the NR iterations to

$$\hat{\Psi}_i(\hat{X}) = \hat{G}\hat{X} + \hat{C}\hat{X} + \hat{F}(\hat{X}) - \hat{B}_t \quad (6.16)$$

with

$$\hat{B}_t = \hat{B}_{dc} - \alpha_i \hat{B}_{ac} \quad (6.17)$$

Starting with an initial guess $\hat{X}_{guess}$, the solution is updated, at each Newton iteration as follows,

$$\hat{X}_{new} = \hat{X}_{old} - J_h(\hat{X}_{old})^{(-1)} \hat{\Psi}_i(\hat{X}_{old}) \quad (6.18)$$

where $\hat{X}_{old}$ is the old guess, $\hat{X}_{new}$ is the new updated solution, and $\hat{J}_h(\hat{X}_{old})$ is the Jacobian of the reduced equations, which can be written as

$$\begin{aligned}
 \hat{J}_h(\hat{X}_{old}) &= \left. \frac{\partial \hat{\Psi}_i(\hat{X})}{\partial \hat{X}} \right|_{\hat{X}_{old}} \\
 &= \hat{G} + \hat{C} + \left. \frac{\partial \hat{F}(\hat{X})}{\partial \hat{X}} \right|_{\hat{X}_{old}} \\
 &= \hat{G} + \hat{C} + \left. \frac{\partial (Q_h^T F(Q_h \hat{X}))}{\partial \hat{X}} \right|_{\hat{X}_{old}} \\
 &= \hat{G} + \hat{C} + Q_h^T \left(\left. \frac{\partial F(X_h)}{\partial X} \right|_{Q_h \hat{X}_{old}} \right) Q_h
 \end{aligned} \tag{6.19}$$

The Jacobian of the reduced circuit is therefore related to the Jacobian of the original system given in Equation (3.71) by a congruent transformation with the matrix Q_h

$$\hat{J}_h = Q_h J_h Q_h \tag{6.20}$$

where $\hat{J}_h \in \mathbb{R}^{N_q \times N_q}$ is much smaller in size than the original Jacobian. The reduced Jacobian therefore be efficiently computed without explicitly evaluating the original large Jacobian.

The iteration in (6.18) is repeated until the difference $\|\Delta \hat{X}\|$ between the old solution $\hat{X}_{old}$ and the updated value is less than the error tolerance ϵ . The analysis is therefore done completely in the domain of the reduced system. Once convergence is achieved the solution is mapped back to the original system space

$$Q_h \hat{X} \rightarrow X \tag{6.21}$$

An estimate of the error can be obtained by substituting the solution X into the Harmonic Balance equations in (6.7), and checking if Kirchhoff's current law is satisfied. If the solution is found to be unsatisfactory, the order of the reduced system can be increased or an expansion at an intermediate value of α can be performed.

6.4 Calculation of the Subspace Q_h

In this section, the algorithm for finding the subspace Q_h is described. The circuit reduction approach which was described in the previous sections is based on an orthogonal projection onto this subspace. Q_h was defined in (6.13) as

$$\text{colsp}[Q_h] = \text{colsp}[M_0, M_1, \dots, M_{N_q-1}]$$

In order to obtain Q_h , the moments $M_0, \dots, M_{N_q-1}$ are evaluated, and a standard orthonormalization method [22] is applied in order to obtain an orthonormal basis Q_h that spans the space of these moments. The focus of this section is therefore on the evaluation of the moments A_k , or in other words the Taylor coefficients of X with respect to the continuation parameter α as defined below

$$X(\alpha) = \sum_{k=0}^{N_q-1} A_k \alpha^k \quad (6.22)$$

Now recall the Harmonic Balance equations appended with the continuation parameter α as defined in (6.7),

$$\Psi(X, \alpha) = \bar{G}X + j\Omega\bar{C}X + F(X) - B_{dc} - \alpha B_{ac} \quad (6.23)$$

The nonlinear vector $F(X)$ can also be expanded in a Taylor series as follows

$$F(X(\alpha)) = \sum_{k=0}^{N_q-1} D_k \alpha^k \quad (6.24)$$

Clearly, at $\alpha = 0$, we have $F(X(0)) = D_0 = F(A_0)$. Substituting from (6.22) and (6.24) into (6.23), we get,

$$\bar{G} \sum_{k=0}^{N_q-1} A_k \alpha^k + j\Omega\bar{C} \sum_{k=0}^{N_q-1} A_k \alpha^k + \sum_{k=0}^{N_q-1} D_k \alpha^k = B_{dc} + \alpha B_{ac} \quad (6.25)$$

Setting $\alpha = 0$ we get

$$\bar{G}A_0 + j\Omega\bar{C}A_0 + F(A_0) = B_{dc} \quad (6.26)$$

Note that A_0 in (6.26) is simply the DC solution. Also, equating the first power of α in (6.25) yields

$$\bar{G}A_1 + j\Omega\bar{C}A_1 + D_1 = B_{ac} \quad (6.27)$$

Now define the matrix $T(\alpha)$ to be

$$T(\alpha) = \frac{\partial F(X)}{\partial X} \quad (6.28)$$

and let $T(\alpha)$ have the Taylor series expansion

$$T(\alpha) = \sum_{k=0} T_k \alpha^k \quad (6.29)$$

Since $D_1 = (\partial F / \partial \alpha)|_{\alpha=0} = (\partial F / \partial X) A_1$, then substituting in (6.27) gives the following relation for A_1

$$(\bar{G} + j\Omega\bar{C} + T_0)A_1 = B_{ac} \quad (6.30)$$

The first moment is therefore found by simply solving the above set of linear equations. Note that the left hand side matrix is simply the Jacobian of the harmonic balance equations. The CPU cost of this operation is one LU decomposition of the Jacobian matrix. In order to compute $A_2, \dots, A_q$, the Taylor series expansion of $T(\alpha)$ is used in the following relation

$\partial F / \partial \alpha$ in the following form:

$$\frac{\partial F}{\partial \alpha} = T \frac{\partial X}{\partial \alpha} \quad (6.31)$$

to get

$$\sum_{i=1} i D_i \alpha^{i-1} = \sum_{i=0} T_i \alpha^i \sum_{i=1} i A_i \alpha^{i-1} \quad (6.32)$$

Taking the n^{th} derivative in (6.32) with respect to α and putting $\alpha = 0$, we get the following recursive relationship

$$D_n = T_0 A_n + \frac{1}{n} \sum_{j=1}^{n-1} (n-j) T_j A_{n-j} \quad (6.33)$$

Equating the n^{th} power of α in (6.25) we get

$$\bar{G}A_n + j\Omega\bar{C}A_n + D_n = 0 \quad (6.34)$$

Now using equation (6.33) the following recursive relationship for the derivatives $\mathbf{A}_n$ is obtained

$$(\bar{\mathbf{G}} + j\Omega\bar{\mathbf{C}} + \mathbf{T}_0)\mathbf{A}_n = -\frac{1}{n} \sum_{j=1}^{n-1} (n-j)\mathbf{T}_j\mathbf{A}_{n-j} \quad (6.35)$$

It is clear from (6.30) and (6.35) that computing $\mathbf{A}_1, \dots, \mathbf{A}_q$ requires only one LU decomposition of the Jacobian matrix. Each additional derivatives can be obtained by forward/backward substitution. Next we proceed to show how to compute the right hand side of equation (6.35). The term $\mathbf{A}_{n-j}$ represents the $(n-j)^{th}$ derivative of $\mathbf{X}$ with respect to α , with $\mathbf{A}_0$ and $\mathbf{A}_{n-1}$ evaluated using (6.30) and (6.35). The higher order derivatives are found recursively using (6.35). The remaining term required in (6.35) is $\mathbf{T}_j$. First, recall the definition of $\mathbf{T}(\alpha)$ from equations (6.28), (6.29) and (3.71)

$$\mathbf{T}(\alpha) = \sum_{j=0} \mathbf{T}_j \alpha_j = \mathbf{P}^T \Xi \begin{bmatrix} g(\mathbf{S}_0 \Xi^{-1} \mathbf{P} \mathbf{X}) \\ \vdots \\ g(\mathbf{S}_{N_h-1} \Xi^{-1} \mathbf{P} \mathbf{X}) \end{bmatrix} \mathbf{S} \Xi^{-1} \mathbf{P} \quad (6.36)$$

where

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_0 \\ \vdots \\ \mathbf{S}_{N_h-1} \end{bmatrix} \quad (6.37)$$

and

$$\mathbf{g}(\mathbf{x}) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \quad (6.38)$$

In (6.36), $\mathbf{S}, \Xi^{-1}$, and $\mathbf{P}$ are independent of α . Finding $\mathbf{T}_j$, therefore, requires expanding $g(\mathbf{S}_i \Xi^{-1} \mathbf{P} \mathbf{X})$ in a Taylor's series,

$$g(\mathbf{S}_i \Xi^{-1} \mathbf{P} \mathbf{X}) = \sum_{j=0} g_j \alpha^j \quad (6.39)$$

Computing $\mathbf{T}_j$ therefore requires computing g_j . Using (6.22), (6.39) becomes

$$\sum_{j=0} g_j \alpha^j = g(\mathbf{S}_i \Xi^{-1} \mathbf{P} \sum_{i=0} \mathbf{A}_i \alpha^i) \quad (6.40)$$

Equation	$y = \sum_{i=0} a_i \alpha^i, z = \sum_{i=0} c_i \alpha^i, f = \sum_{i=0} d_i \alpha^i$
$f = \exp(y)$	$d_0 = \exp(a_0)$ $d_n = (1/n) \sum_{i=0}^{n-1} d_i a_{n-i} (n-i)$
$f = \log(y)$	$d_0 = \log(a_0)$ $d_n = (1/a_0) \left(a_n - \sum_{i=1}^{n-1} a_i d_{n-i} \frac{(n-i)}{n} \right)$
$f = y^p$	$d_0 = a_0^p$ $d_n = (1/na_0) \left(npd_0 a_n + \sum_{i=1}^{n-1} d_i a_{n-i} (pn - i(p+1)) \right)$
$f = y + z$	$d_n = a_n + c_n$
$f = y - z$	$d_n = a_n - c_n$
$f = yz$	$d_n = \sum_{i=0}^n a_i c_{n-i}$
$f = y/z$	$d_n = (1/c_0) \left(a_n - \sum_{i=0}^{n-1} a_i c_{n-i} \right)$

Table 6.1: Formulas for the derivatives of some simple functions [97]

The problem of computing g_j based on the knowledge of $A_0, \dots, A_{i-1}$ has been addressed in [97]. As an example for this computation, consider the case where

$$f(x) = I_s(e^{x/V_t} - 1); \quad g(x) = \frac{I_s}{V_t} e^{x/V_t} = \sum_{i=0} g_i \alpha^i \quad (6.41)$$

Let x be represented by the Taylor's series

$$x = \sum_{i=0} a_i \alpha^i \quad (6.42)$$

It can be shown that

$$g_n = \frac{I_s}{nV_t} x = \sum_{i=0}^{n-1} g_i \alpha^{n-i} (n-i) \quad (6.43)$$

Similar expressions can be found for other functions as shown in Table 6.1.

6.5 Computational Complexity

There are two main sources of CPU computational cost in the proposed circuit reduction algorithm. The first CPU computational effort is spent in the calculation of the moments and the second source of CPU cost results from the construction and solution of the reduced system.

6.5.1 Moments Computation

To illustrate the computational cost involved in the moments computation process, consider the equations of the auxiliary system

$$\bar{G}X + j\Omega\bar{C}X + F(X) = B_{dc} + \alpha B_{ac} \quad (6.44)$$

and rewrite them in the time-domain to obtain

$$Gx(t, \alpha) + C\dot{x}(t, \alpha) + f(x(t, \alpha)) = b_{dc} + \alpha b_{ac}(t) \quad (6.45)$$

where the dependence of the waveform on the parameter α is stated explicitly in (6.45) by denoting the circuit variables vector x as a function of two variables: t and α . Setting $\alpha = 0$ in (6.45) yields

$$Gx(t, 0) + C\dot{x}(t, 0) + f(x(t, 0)) = b_{dc} \quad (6.46)$$

Since the input sources contained in the vector b_{dc} in (6.46) represent the DC biasing sources, then the response $x(t, 0)$ represents the DC operating point and is therefore independent of t and may be written as

$$x(t, 0) = x_{dc} \quad (6.47)$$

where x_{dc} denotes the DC operating point. Taking the derivatives of both sides of (6.45) w.r.t. α gives

$$G \frac{\partial x(t, \alpha)}{\partial \alpha} + C \frac{\partial}{\partial t} \left(\frac{\partial x(t, \alpha)}{\partial \alpha} \right) + g(x(t, \alpha)) \frac{\partial x(t, \alpha)}{\partial \alpha} = b_{ac}(t) \quad (6.48)$$

where

$$\mathbf{g}(\mathbf{x}(t, \alpha)) = \frac{\partial \mathbf{f}(\mathbf{x}(t, \alpha))}{\partial \mathbf{x}} \quad (6.49)$$

Let both $\mathbf{x}(t, \alpha)$ and $\mathbf{g}(\mathbf{x}(t, \alpha))$ be expanded in Taylor's series w.r.t. α as in

$$\begin{aligned} \mathbf{x}(t, \alpha) &= \sum_{k=0} \mathbf{a}_k(t) \alpha^k \\ \mathbf{g}(\mathbf{x}(t, \alpha)) &= \sum_{k=0} \mathbf{b}_k(t) \alpha^k \end{aligned} \quad (6.50)$$

where $\mathbf{a}_k(t)$ and $\mathbf{b}_k(t)$ represent, respectively, the time-dependent moments of $\mathbf{x}(t, \alpha)$ and $\mathbf{g}(\mathbf{x}(t, \alpha))$ computed at $\alpha = 0$. Simple mathematical manipulation shows that $\mathbf{b}_0(t)$ must be independent of t since

$$\begin{aligned} \mathbf{b}_0(t) &= \mathbf{g}(\mathbf{x}(t, 0)) \\ &= \mathbf{g}(\mathbf{x}_{dc}) \\ &\equiv \mathbf{g}_{dc} \end{aligned} \quad (6.51)$$

Substituting from (6.50) and (6.51) into (6.48) yields

$$\mathbf{G} \sum_{k=1} k \mathbf{a}_k(t) \alpha^{k-1} + \mathbf{C} \sum_{k=1} k \frac{d\mathbf{a}_k(t)}{dt} \alpha^{k-1} + \sum_{i=0} \sum_{j=1} j \mathbf{b}_i(t) \mathbf{a}_j(t) \alpha^{i+j-1} = \mathbf{b}_{ac}(t) \quad (6.52)$$

Setting $\alpha = 0$ in (6.52) results in

$$\mathbf{G} \mathbf{a}_1(t) + \mathbf{C} \frac{d\mathbf{a}_1(t)}{dt} + \mathbf{g}_{dc} \mathbf{a}_1(t) = \mathbf{b}_{ac}(t) \quad (6.53)$$

By repeatedly taking the derivative of (6.52) w.r.t. α and setting $\alpha = 0$, we obtain the following systems of equations

$$\begin{aligned} \mathbf{G} \mathbf{a}_2(t) + \mathbf{C} \frac{d\mathbf{a}_2(t)}{dt} + \mathbf{g}_{dc} \mathbf{a}_2(t) &= -\frac{1}{2} \mathbf{b}_1(t) \mathbf{a}_1(t) \\ \mathbf{G} \mathbf{a}_3(t) + \mathbf{C} \frac{d\mathbf{a}_3(t)}{dt} + \mathbf{g}_{dc} \mathbf{a}_3(t) &= -\frac{1}{3} [2\mathbf{b}_2(t) \mathbf{a}_1(t) + \mathbf{b}_1(t) \mathbf{a}_2(t)] \\ &\vdots \\ \mathbf{G} \mathbf{a}_n(t) + \mathbf{C} \frac{d\mathbf{a}_n(t)}{dt} + \mathbf{g}_{dc} \mathbf{a}_n(t) &= -\frac{1}{n} \sum_{j=1}^{n-1} (n-j) \mathbf{b}_{n-j}(t) \mathbf{a}_j(t) \end{aligned} \quad (6.54)$$

Equations (6.54) shows that n^{th} moment (or scaled derivative) of $x(t, \alpha)$ w.r.t. α at $\alpha = 0$ is simply the response of a Linear Time-Invariant (LTI) system whose input signal is dependent on the previous moments. This LTI system is obtained by linearizing the nonlinear equation around the DC operating point. Figure 6.2 shows a schematic representation for the setting of Equation (6.54).

Figure 6.2: Schematic representation for computing the moments in (6.54).

It is important to note here that computing the moments through applying the HB techniques on Equations (6.54) is mathematically equivalent to the computation explained in Section 6.4 through (6.30) and (6.35). Using the time-domain expressions of (6.54), however, emphasizes the fact that computing the moments is computationally equivalent to applying the HB technique to a *linear* system repetitively using the output from the previous phases as an

input to the next phase. Section 3.2.1 showed that applying the HB approach to linear circuits in order to find the steady-state response is computationally inexpensive as the problem can be reduced to a set of a decoupled *sparse* systems of equations as shown in Equation (3.24). Each one of those systems is corresponding to a multiple of the fundamental frequency and its size is equal to the size of the MNA formulation N .

6.5.2 Construction of the Reduced Jacobian

The main CPU cost of solving the reduced system is consumed in constructing the reduced Jacobian matrix. This part can be made more efficient by exploiting the structure of the Jacobian matrix in the original system. For that purpose, consider the expression of the reduced Jacobian matrix from Equation (6.19)

$$\hat{J}_h = \hat{G} + \hat{C} + Q_h^T \left(\frac{\partial F(X)}{\partial X} \bigg|_{Q_h \hat{X}_{old}} \right) Q_h \quad (6.55)$$

The first two terms in (6.55) can be computed once at $\alpha = 0$ and stored during the tracing of $\alpha : 0 \rightarrow 1$. However, it is the third term that varies with α and must be computed during the tracking process. Computing the third term can be made efficient using the structure of $\frac{\partial F(X)}{\partial X}$ as shown in (3.71):

$$Q_h^T \underbrace{[P^T \Xi \mathcal{D} \bar{f} \Xi^{-1} P]}_{\Lambda} Q_h \quad (6.56)$$

Computing (6.56) can be achieved by calculating the effect of the term in the square brackets, Λ , on the columns of the matrix Q_h which can be carried out using IFFT, MVP, then FFT. It has been shown in [28] that computing the effect of Λ on the columns of Q_h using FFT and IFFT is computationally more efficient than computing Λ explicitly then using MVP.

Computing the effect of Λ on a vector is done by first rearranging the operand vector, using the permutation matrix to put it in node-major indexing mode, then applying IFFT on the permuted vector then effecting an MVP with the matrix $\mathcal{D} \bar{f}$ defined in Equation (3.72). $\mathcal{D} \bar{f}$

Figure 6.3: Sparsity pattern of $\mathcal{D}\bar{f}$.

is typically a sparse block matrix consisting of diagonal blocks with a typical sparsity pattern similar to the one shown in Figure 6.3. The resulting vector is then subjected to an FFT process and then permuted again to put it in node-minor indexing mode. The computational cost of performing this operation is $O(NN_h \log_2(N_h))$ for N circuit variables and N_h sampling time points.

One advantage of the proposed circuit reduction algorithm is that it lends itself easily to parallel implementation since the process of computing the effect of Λ on the columns of Q_h can be accomplished in parallel.

6.6 Proof of Preservation of Derivatives

One of the main features of the reduced order system presented in this chapter is that it shares the first q derivatives with respect to α with the original system. In this section we prove that

the reduced system in (6.10) shares the first N_q derivatives with the original system in (6.7) as stated in (6.15). The proof is given by *mathematical induction*. First, it shall be proven that the zeroth derivative A_0 is preserved. Then we show that the first derivative $\hat{A}_1$ of $\hat{X}$ with respect to α as obtained from the reduced equations in (6.10) is equivalent to the one obtained from the original HB equations in (6.7) through mapping by Q_h . Next we proceed to show that the same relation applies to the k^{th} derivative of $\hat{X}$ provided that the previous $(k - 1)$ derivatives are preserved under the mapping Q_h . In other words, for $k = 0, \dots, N_q - 1$, if A_k is the k^{th} derivative of the original system, and $\hat{A}_k$ is that of the reduced system then we have

$$A_k = Q_h \hat{A}_k \quad (6.57)$$

First $\hat{X}(\alpha)$ and $F(Q_h \hat{X})$ in (6.9) are expanded as Taylor series in the form

$$\hat{X}(\alpha) = \sum_{k=0} \hat{A}_k \alpha^k \quad (6.58a)$$

$$F(Q_h \hat{X}) = \sum_{k=0} \hat{D}_k \alpha^k \quad (6.58b)$$

Substituting from (6.58) into (6.10) results in the following

$$(\hat{G} + \hat{C}) \sum_{k=0} \hat{A}_k \alpha^k + Q_h^T \sum_{k=0} \hat{D}_k \alpha^k - Q_h^T B_{dc} - \alpha Q_h^T B_{ac} = 0 \quad (6.59)$$

Equating similar powers of α in equation (6.59) results in

$$(\hat{G} + \hat{C}) \hat{A}_0 + Q_h^T \hat{D}_0 = Q_h^T B_{dc} \quad (6.60a)$$

$$(\hat{G} + \hat{C}) \hat{A}_1 + Q_h^T \hat{D}_1 = Q_h^T B_{ac} \quad (6.60b)$$

$$(\hat{G} + \hat{C}) \hat{A}_k + Q_h^T \hat{D}_k = 0 \quad (6.60c)$$

Proof of Preservation of A_0

In this section it shall be proven that the zeroth moment A_0 is preserved in the reduced system.

Noting that

$$\hat{D}_0 = F(Q_h \hat{A}_0) \quad (6.61)$$

Equation (6.60a) can be written as

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) Q_h \hat{A}_0 + F(Q_h \hat{A}_0) - B_{dc} \right) = 0 \quad (6.62)$$

Since Q_h is an orthonormal basis for the subspace spanned by the columns of the matrix $[A_0, \dots, A_{N_q-1}]$, it can be represented in the form [22]

$$Q_h = [A_0, \dots, A_{N_q-1}] R^{-1} \quad (6.63)$$

where R is an upper triangular matrix. Substituting from (6.63) into (6.62) results in

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) [A_0, \dots, A_{N_q-1}] R^{-1} \hat{A}_0 + F([A_0, \dots, A_{N_q-1}] R^{-1} \hat{A}_0) - B_{dc} \right) = 0 \quad (6.64)$$

Now we verify that $e_1 = R^{-1} \hat{A}_0$ satisfies equation (6.64), where e_r represents the r^{th} column of the identity matrix. Substituting into (6.64) yields

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) [A_0, \dots, A_{N_q-1}] e_1 + F([A_0, \dots, A_{N_q-1}] e_1) - B_{dc} \right) = 0 \quad (6.65)$$

Simplifying equation (6.65) results in

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) A_0 + F(A_0) - B_{dc} \right) = 0 \quad (6.66)$$

Comparing with (6.26) it is clear that Equation (6.66) is satisfied. Therefore we can conclude that $e_1 = R^{-1} \hat{A}_0$ is indeed a solution of equation (6.64). $\hat{A}_0$ can then be written as

$$\hat{A}_0 = R e_1 \quad (6.67)$$

Pre-multiplying by Q_h results in

$$Q_h \hat{A}_0 = Q_h R e_1 = [A_0, \dots, A_{N_q-1}] R^{-1} R e_1 = [A_0, \dots, A_{N_q-1}] e_1 = A_0 \quad (6.68)$$

Therefore the zeroth moment A_0 is preserved under the mapping Q_h .

Proof of Preservation of A_1

From equation (6.58b) $\hat{D}_1$ can be written as

$$\hat{D}_1 = \left. \frac{\partial F(Q_h \hat{X})}{\partial \alpha} \right|_{\alpha=0} = \left. \frac{\partial F(Q_h \hat{X})}{\partial (Q_h \hat{X})} \frac{\partial Q_h \hat{X}}{\partial \hat{X}} \frac{\partial \hat{X}}{\partial \alpha} \right|_{\alpha=0} \quad (6.69)$$

Using the definition in (6.28), (6.69) reduces to

$$\hat{D}_1 = T_0 Q_h \hat{A}_1 \quad (6.70)$$

Substituting (6.70) into (6.60b), we get

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) Q_h \hat{A}_1 + T_0 Q_h \hat{A}_1 - B_{ac} \right) = 0 \quad (6.71)$$

Next we substitute the value for Q_h from (6.63) to get

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) [A_0, \dots, A_{N_q-1}] R^{-1} \hat{A}_1 + T_0 [A_0, \dots, A_{N_q-1}] R^{-1} \hat{A}_1 - B_{ac} \right) = 0 \quad (6.72)$$

Now we verify that $e_2 = R^{-1} \hat{A}_1$ satisfies equation (6.64), where e_r represents the r^{th} column of the identity matrix. Substituting into (6.72) results in

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) [A_0, \dots, A_{N_q-1}] e_2 + T_0 [A_0, \dots, A_{N_q-1}] e_2 - B_{ac} \right) = 0 \quad (6.73)$$

Simplifying (6.73) we obtain the following

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) A_1 + T_0 A_1 - B_{ac} \right) = 0 \quad (6.74)$$

Comparing with (6.30), it is clear that equation (6.74) is satisfied. The first moment $\hat{A}_1$ of the reduced system can therefore be written as

$$\hat{A}_1 = R e_2 \quad (6.75)$$

Pre-multiplying by Q_h results in

$$Q_h \hat{A}_1 = Q_h R e_2 = [A_0, \dots, A_{N_q-1}] R^{-1} R e_2 = [A_0, \dots, A_{N_q-1}] e_2 = A_1 \quad (6.76)$$

Therefore the zeroth moment A_1 is preserved under the mapping Q_h .

Proof of Preservation of A_n

In this section it shall be proven that the n^{th} moment A_n is preserved in the reduced system if the previous moments $A_0, \dots, A_{n-1}$ are also preserved. First the matrix $\hat{T}$ is defined as

$$\hat{T}(\alpha) = \frac{\partial F(Q_h \hat{X})}{\partial \hat{X}} \quad (6.77)$$

The Taylor expansion of $\hat{T}$ with respect to α is

$$\hat{T}(\alpha) = \sum_{k=0} \hat{T}_k \alpha^k \quad (6.78)$$

Recalling that $\hat{A}_k$ and $\hat{D}_k$ are the Taylor coefficients of $\hat{X}$ and $F(Q_h \hat{X})$ respectively as defined in equations (6.58a) and (6.58b), and using the chain rule results in

$$\frac{\partial F(Q_h \hat{X})}{\partial \alpha} = \frac{\partial F(Q_h \hat{X})}{\partial \hat{X}} \frac{\partial \hat{X}}{\partial \alpha} = \hat{T}(\alpha) \frac{\partial \hat{X}}{\partial \alpha} \quad (6.79)$$

or

$$\sum_{k=1} k \hat{D}_k \alpha^{k-1} = \sum_{k=0} \hat{T}_k \alpha^k \sum_{k=1} k \hat{A}_k \alpha^{k-1} \quad (6.80)$$

Equating the coefficients of equal powers of α in (6.79) results in the following recursive relationship

$$\hat{D}_k = \hat{T}_0 \hat{A}_k + \frac{1}{k} \sum_{j=1}^{k-1} (k-j) \hat{T}_j \hat{A}_{k-j} \quad (6.81)$$

Using the chain rule, $\hat{T}$ can be related to the matrix T which was defined in (6.28)

$$\hat{T} = \frac{\partial F(Q_h \hat{X})}{\partial \hat{X}} = \frac{\partial F(Q_h \hat{X})}{\partial (Q_h \hat{X})} \frac{\partial (Q_h \hat{X})}{\partial \hat{X}} = T Q_h = \sum_{k=0} T_k Q_h \alpha^k \quad (6.82)$$

Substituting (6.82) into (6.81) results in

$$\hat{D}_k = T_0 Q_h \hat{A}_k + \frac{1}{k} \sum_{j=1}^{k-1} (k-j) T_j Q_h \hat{A}_{k-j} \quad (6.83)$$

The recursive equation in (6.60c) for finding the n^{th} moment of the reduced system can therefore be written as

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) Q_h \hat{A}_k + T_0 Q_h \hat{A}_k + \frac{1}{k} \sum_{j=1}^{k-1} (k-j) T_j Q_h \hat{A}_{k-j} \right) = 0 \quad (6.84)$$

Substituting for Q_h from (6.63) results in

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) [A_0, \dots, A_{N_q-1}] R^{-1} \hat{A}_k + T_0 [A_0, \dots, A_{N_q-1}] R^{-1} \hat{A}_k + \frac{1}{k} \sum_{j=1}^{k-1} (k-j) T_j Q_h \hat{A}_{k-j} \right) = 0 \quad (6.85)$$

Substituting $e_{k+1} = R^{-1} \hat{A}_k$ as a solution for the above equation results in

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) [A_0, \dots, A_{N_q-1}] e_{k+1} + T_0 [A_0, \dots, A_{N_q-1}] e_{k+1} + \frac{1}{k} \sum_{j=1}^{k-1} (k-j) T_j Q_h \hat{A}_{k-j} \right) = 0 \quad (6.86)$$

Assuming that the previous moments $A_0, \dots, A_{k-1}$ are preserved under the mapping Q_h or more precisely

$$A_i = Q_h \hat{A}_i; \quad i = 0, \dots, k-1 \quad (6.87)$$

Equation (6.86) can be written as

$$Q_h^T \left((\bar{G} + j\Omega \bar{C}) A_k + T_0 A_k + \frac{1}{k} \sum_{j=1}^{k-1} (k-j) T_j A_{k-j} \right) = 0 \quad (6.88)$$

Comparing with (6.35), it is clear that equation (6.88) is satisfied. The k^{th} moment $\hat{A}_k$ of the reduced system can therefore be written as

$$\hat{A}_k = R e_{k+1} \quad (6.89)$$

Pre-multiplying by Q_h results in

$$Q_h \hat{A}_k = Q_h R e_{k+1} = [A_0, \dots, A_{N_q-1}] R^{-1} R e_{k+1} = [A_0, \dots, A_{N_q-1}] e_{k+1} = A_k \quad (6.90)$$

Therefore the zeroth moment A_k is preserved under the mapping Q_h if the previous moments $A_0, \dots, A_{k-1}$ are preserved. It can be therefore deduced by induction that the reduced system preserves the first N_q moments of the original system under the mapping Q_h , where N_q is the number of column moments forming the subspace spanned by Q_h .

6.7 Numerical Examples

6.7.1 Example 1

The proposed circuit reduction technique was applied to a bipolar transistor amplifier circuit. The congruent transformation matrix was obtained from the derivatives at the DC solution, requiring only one L/U factorization of the Jacobian matrix. The harmonic balance algorithm was then applied to the reduced circuit and the solution was compared with that of the original circuit. The output response as a function of the amplitude of the input waveform is shown in Figures 6.4, 6.5, and 6.6. As can be seen from the plots, a congruent transformation of order 16 is enough to match the response of the original circuit up to high distortion levels. The size of the Jacobian of the reduced circuit was only 16x16 compared with that of the original circuit Jacobian of 429x429. The proposed algorithm required only one solution of the large jacobian equations. Conventional analysis however, required this solution 63 times (one for each Newton iteration at each step of the continuation method). Figure 6.7 shows a comparison

Figure 6.4: Example 1: Solution at the first harmonic.

for the time-domain as obtained through solution of the original circuit using classical HB and the solution obtained through three different reduced circuits.

Figure 6.5: Example 1: Solution at the third harmonic.

Figure 6.6: Example 1: Solution at the fifth harmonic.

Figure 6.7: Periodic time-domain response of the bipolar circuit (Example 1).

Figure 6.8: Example 2: Comparison of intermodulation tones.

6.7.2 Example 2

An amplifier circuit was considered with two input tones at 170KHz and 180KHz. The amplitude of the input signal was 0.25V for both tones. The size of the harmonic balance equations was $N = 793$. The moments of the system were obtained for both tones simultaneously, and the reduced order system was of order $q = 25$. In Figure 6.8, the amplitude of the output at the fundamental frequencies 170KHz and 180KHz and some of the intermodulation tones are compared. The results from the reduced system matched those of the original system. Note that for the reduced system a Jacobian of size 25×25 was inverted at each Newton iteration, while the Jacobian of the original system was of size 793×793 .

6.7.3 Example 3

An Op-Amp circuit consisting of 5 transistors is considered for this example (Figure 6.9). Conventional HB analysis required 24 harmonics to approximate the steady state response for this circuit. Performing the HB analysis needed 67 NR iterations with 67 LU decompositions of a Jacobian matrix of size 2400x2400. Using the proposed algorithm, a reduced set of nonlinear equations of order 35 was constructed using the derivatives at the DC operating point. The solution of the reduced set of nonlinear equations was then tracked up to the full swing of the input power. Figures 6.10, 6.11, and 6.12 depict the behavior of the 2nd, 5th and 9th harmonics of the output node voltage in both the original and reduced circuits versus the continuation parameter α . As can be seen the behavior of the harmonics in the reduced system is indistinguishable from that of the original system. Figure 6.13 shows a comparison for the magnitudes of all the harmonics at the output node as computed from both the original and reduced circuit at the terminus of the continuation sweep. The CPU cost was measured for both conventional harmonic balance simulation and the proposed method, and a speedup factor of 12.3 was achieved without loss of accuracy.

Figure 6.9: OpAmp circuit of example 3.

Figure 6.10: Example 3: Amplitude of the second harmonic.

Figure 6.11: Example 3: Amplitude of the fifth harmonic.

Figure 6.12: Example 3: Amplitude of the 9th harmonic.

Figure 6.13: Example 3: Harmonics at terminus of continuation sweep.

Chapter 7

MOR of Linear Periodically Time Varying Systems

7.1 Introduction

Abstraction of large communication blocks with small macromodels that preserve their basic features is essential for efficient hierarchical system-level simulation. Generating small accurate reduced-order models that capture the behavior of much bigger systems has proven to be key in developing successful simulation algorithms.

In recent years, model-reduction of Linear Time-Invariant (LTI) systems has been well-established and successful for efficient simulation of large linear circuits [1], [92], [15], [98]. The idea that large LTI systems can be replaced with systems of a much smaller size stems from the fact that the response of an LTI system is dominated by a few number of eigenvalues of the coefficient matrix. In circuit applications, however, there are many other systems that can not be modelled as LTI systems. Consider, for example, an RF component of a receiver subsystem. This component is designed to have an almost linear response w.r.t. the small-signal coming from the RF source. On the other hand, it exhibits a strong nonlinear behavior

for the large local oscillator signal. Such a system can be modelled rather as a Linear Time-Varying (LTV) system as far as the small signal is concerned. In this case, the LTV system is obtained in two steps. In the first step, the response of the system is computed due to the strong signal assuming that the small signal is vanishing. If the large signal happens to be periodic, then steady-state approaches can be used to compute this solution [17], [99], [56]. In the second step, the solution obtained is used to linearize the circuit around its trajectory, where the system in this case appears to be linear w.r.t. the small signal but with time varying coefficients [100].

Unfortunately, LTI model reduction techniques are not adequate to capture behavioral characteristics of LTV systems due to the fact that the latter ones possess some unique features (e.g. frequency translation). The problem of generating reduced-order LTV macromodels for circuit applications has been addressed recently, to the best of the authors' knowledge, in [12] where the approach proposed therein is based on the concept of Time-Varying System function (TVS) introduced by Zadeh [31] to describe time-varying networks. In particular, for circuit applications, LTV systems are periodically changing with time and can be described by TVS function, $h(s, t)$, that is periodical in t . This fact enables steady-state approaches to be invoked to transform the system of *differential* equations describing $h(s, t)$ into a system *algebraic* equations. Two approaches based on frequency- and time-domain formulations were suggested to achieve that; the one based on frequency-domain approach uses the idea of the Harmonic Balance (HB) while the one based on the time-domain uses a collocation-based Finite-Differences (FD) method. The advantage of dealing with algebraic systems though, is that LTI model reduction techniques are easily applied to an algebraic system of equations.

In fact, this approach represents a significant step in that a reduced-order Linear Periodically Time-Varying (LPTV) macromodels can be constructed to represent large LPTV systems. However, it requires a solution of large system of equations that arise in the inner loop of the HB or FD difference methods.

This chapter presents a new approach for generating LPTV reduced-order macromodels that does not require solution of large system of equations. The proposed algorithm presents a two-fold (theoretical and computational) contribution. The theoretical aspect addresses the problem of approximating a TVS function that has its time-dependent derivatives w.r.t. the frequency variable s related recursively through a differential operator. An algorithm for constructing a time-varying orthogonal basis for the derivatives (or moments) without computing the moments explicitly is given. This orthogonalization algorithm represents the natural counterpart of the classical Arnoldi process used to orthogonalize a set of *time-independent* moments related through an *algebraic* operator. The construction of the orthonormal basis without computing the moments explicitly is crucial to having accurate reduced-order LPTV macromodels. We show that using this orthonormal basis in an integrated congruence transform to reduce the system equations generates accurate reduced-order LPTV macromodels.

On the computational side, the proposed algorithm does not require solving a huge system of equations. In fact the main computational advantage is motivated by the fact that the TVS function, $h(s, t)$, has very short transient response for values of s that are in the right-half plane of the s -domain. This means that the time-varying moments of $h(s, t)$ w.r.t. s take a very short time before settling into steady state, and hence generating the orthonormal basis can be carried out efficiently by expanding $h(s, t)$ in the right-half plane and performing time-domain integration [33] for a very short period of time. In addition, we present an efficient procedure to eliminate numerical noise resulting from the truncation error generated by the time-domain integration method. The proposed algorithm shows one-to-two order of magnitude speed-up over the algebraic-based approach.

The rest of the chapter is organized as follows. In Section 7.2 we present a brief background on LTV systems and the concept of time-varying transfer function. Section 7.3 presents a quick review of the recent approaches to model-order reduction of LPTV systems. Section 7.4 describes the development of the proposed LPTV macromodelling algorithm. Section 7.5

presents the orthonormal basis construction algorithm. Sections 7.6 and 7.7 present some numerical results and conclusion.

7.2 Time-Varying System Function

A nonlinear system driven by a large signal $\mathbf{b}_L(t)$ and a small signal $u(t)$ can be described by a set of differential-algebraic nonlinear equations,

$$\begin{aligned} \frac{d\mathbf{q}(\mathbf{x}(t))}{dt} + \mathbf{f}(\mathbf{x}(t)) &= \mathbf{b}_L(t) + \mathbf{b}u(t) \\ \mathbf{z}(t) &= \mathbf{d}^T \mathbf{x}(t) \end{aligned} \quad (7.1)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the vector of node voltages; $\mathbf{q}(\cdot)$ and $\mathbf{f}(\cdot)$ are nonlinear functions describing the charge/flux and resistive elements, respectively; $\mathbf{z}(t)$ is the set of output nodes; $\mathbf{b}$ and $\mathbf{d}$ are selector vectors that map the input and output ports to the space of the network nodes, and n is the number of nodes in the network. Splitting the response into two parts:

$$\mathbf{x}(t) = \mathbf{x}_L(t) + \mathbf{x}_s(t) \quad (7.2)$$

where $\mathbf{x}_L(t)$ and $\mathbf{x}_s(t)$ are the responses due to the large signal $\mathbf{b}_L(t)$ and small signal $u(t)$, respectively. Substituting from (7.2) into (7.1) and linearizing around $\mathbf{x}_L(t)$ yields the LTV system,

$$\frac{d(\mathbf{C}(t)\mathbf{x}_s(t))}{dt} + \mathbf{G}(t)\mathbf{x}_s(t) = \mathbf{b}u(t) \quad (7.3)$$

where

$$\mathbf{G}(t) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_L(t)} \quad \mathbf{C}(t) = \left. \frac{\partial \mathbf{q}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_L(t)} \quad (7.4)$$

Zadeh [31] has introduced the concept of time-varying system (transfer) function to describe the response of LTV systems. According to Zadeh's formalism, an LTV system can be characterized by using a time-varying transfer function, $\mathbf{h}(s, t)$, where the response of the

system due to an input of the form e^{st} is given by $e^{st}\mathbf{h}(s, t)$ [32]. Thus, by substituting in (7.3) we get,

$$\frac{d(\mathbf{C}(t)\mathbf{h}(s, t))}{dt} + s\mathbf{C}(t)\mathbf{h}(s, t) + \mathbf{G}(t)\mathbf{h}(s, t) = \mathbf{b} \quad (7.5)$$

The transfer path from the input to the set of the output nodes can be represented by a time-varying transfer function $\Phi(s, t)$ where,

$$\Phi(s, t) = \mathbf{d}^T \mathbf{h}(s, t) \quad (7.6)$$

It is clear from (7.6) that a full characterization of the system transfer function requires a solution of LTV system for many values of s .

In the context of circuit simulation the matrices $\mathbf{C}(t)$ and $\mathbf{G}(t)$ are periodic [100], and thus $\mathbf{h}(s, t)$ (and hence $\Phi(s, t)$) become periodic. In other words, $\mathbf{h}(s, t)$ can be represented as an infinite Fourier series, i.e.,

$$\mathbf{h}(s, t) = \sum_{i=-\infty}^{\infty} H_i(s) e^{j\omega_0 t} \quad (7.7)$$

where $\omega_0 = 2\pi/T$ and T is the period of the system oscillation. Thus the response of the system in (7.3) to a sinusoidal input of frequency $s = j\omega_s$ can be written as

$$v_{\text{out}} = \sum_{i=-\infty}^{\infty} H_i(s) e^{j(\omega_0 + \omega_s)t}.$$

Figure 7.1 depicts the frequency-domain response of a typical LPTV system.

An Example of Time-Varying Transfer Function

To illustrate the concept of time-varying transfer function consider the circuit shown in Figure 7.2. This circuit consists of a low-pass filter followed by an ideal mixer and two band-pass filters. It can be seen here that the transfer path of a signal of the form e^{st} can be described by the following transfer function

$$h(s, t) = H_{-1}(s) e^{-j\omega_0 t} + H_1(s) e^{j\omega_0 t} \quad (7.8)$$

 Figure 7.1: Typical response of a periodically time-varying system of frequency ω_0

 Figure 7.2: Low-Pass filter $\rightarrow$ mixer $\rightarrow$ two bandpass filters

where

$$H_1(s) = \frac{0.5}{1 + sR_1C_1} \frac{\frac{(s+j\omega_0)L_2}{1+(s+j\omega_0)^2L_2C_2}}{R_2 + \frac{(s+j\omega_0)L_2}{1+(s+j\omega_0)^2L_2C_2}} \frac{\frac{(s+j\omega_0)L_3}{1+(s+j\omega_0)^2L_3C_3}}{R_3 + \frac{(s+j\omega_0)L_3}{1+(s+j\omega_0)^2L_3C_3}} \quad (7.9)$$

and

$$H_{-1}(s) = \frac{0.5}{1 + sR_1C_1} \frac{\frac{(s-j\omega_0)L_2}{1+(s-j\omega_0)^2L_2C_2}}{R_2 + \frac{(s-j\omega_0)L_2}{1+(s-j\omega_0)^2L_2C_2}} \frac{\frac{(s-j\omega_0)L_3}{1+(s-j\omega_0)^2L_3C_3}}{R_3 + \frac{(s-j\omega_0)L_3}{1+(s-j\omega_0)^2L_3C_3}} \quad (7.10)$$

Note that $H_{-1}(s) = H_1^*(-s)$.¹ However, it should be noted that in general $H_i(s) \neq H_i^*(-s)$ as in the case of transfer functions of LTI systems. For example, $|H_1(s)|$ of (7.9) is not symmetric w.r.t. to s as shown in Figure 7.3. Figure 7.4 illustrates using $H_1(s)$ and $H_{-1}(s)$ in constructing the frequency-domain spectrum of the output of the mixer circuit of Figure 7.2.

¹In general $H_{-i}(s) = H_i^*(-s)$ for any LPTV system.

Figure 7.3: $|H_1(s)|$ for the circuit of Figure 7.2. ($R_1 = 160\Omega$, $R_2 = 1.6k\Omega$, $C_1 = C_2 = C_3 = 10nF$, $L_2 = L_3 = 25.35nH$, $\omega_0 = 10MHz$)

7.3 Review of MOR for LPTV Systems

Computing the time-varying path of signals through solving LTV system for various values of frequency s can be very expensive. Model-reduction of LTV systems was proposed to characterize transfer characteristics of such systems efficiently [12]. A quick review of this approach is presented here to facilitate understanding the basic ideas in the proposed algorithm. The task of generating small LTV macromodels is essentially constructing reduced-order approximation for $\Phi(s, t)$. In particular, if the matrices $C(t)$ and $G(t)$ are periodic, as it is always the case in circuit applications, then $h(s, t)$ (and hence $\Phi(s, t)$) becomes periodic in t , and classical steady-state approaches can be invoked to obtain compact representations for it. This is discussed in the next subsections.

Figure 7.4: Frequency response of the bandpass mixer of Figure 7.2.

7.3.1 Frequency-Domain Matrix Form

A frequency-domain representation is obtained by representing all time-varying terms in (7.5) in Fourier series. Expanding $C(t)$, $G(t)$, and $h(s, t)$ as Fourier series,

$$\begin{aligned} C(t) &= \sum_{i=-\infty}^{\infty} C_i e^{j\omega_0 t} \\ G(t) &= \sum_{i=-\infty}^{\infty} G_i e^{j\omega_0 t} \\ h(s, t) &= \sum_{i=-\infty}^{\infty} H_i(s) e^{j\omega_0 t} \end{aligned} \quad (7.11)$$

Substituting in (7.5) we get the following system of algebraic equations

$$[s\mathcal{C}_{FD} + (\mathcal{G}_{FD} + \Omega\mathcal{C}_{FD})] \mathbf{H}_{FD}(s) = \mathbf{B}_{FD} \quad (7.12)$$

where

$$\mathcal{C}_{FD} = \begin{pmatrix} \vdots & \vdots & \vdots \\ \cdots & C_0 & C_{-1} & C_{-2} & \cdots \\ \cdots & C_1 & C_0 & C_{-1} & \cdots \\ \cdots & C_2 & C_1 & C_0 & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix} \quad (7.13)$$

$$\mathcal{G}_{FD} = \begin{pmatrix} \vdots & \vdots & \vdots \\ \cdots & G_0 & G_{-1} & G_{-2} & \cdots \\ \cdots & G_1 & G_0 & G_{-1} & \cdots \\ \cdots & G_2 & G_1 & G_0 & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix} \quad (7.14)$$

$$\mathbf{H}_{FD}(s) = [\cdots, H_{-1}(s), H_0(s), H_1(s), \cdots]$$

$$\Omega = j\omega_0 \begin{pmatrix} \ddots & & & & & \\ & -2I & & & & \\ & & -I & & & \\ & & & 0 & & \\ & & & & I & \\ & & & & & 2I \\ & & & & & & \ddots \end{pmatrix} \quad (7.15)$$

$$\mathbf{B}_{FD} = [\cdots, 0, \mathbf{b}^T, 0, \cdots]^T \quad (7.16)$$

7.3.2 Time-Domain Matrix Form

A time-domain version of the equations in (7.12) can be constructed by collocating $\mathbf{h}(s, t)$ over time samples $t \in [0, T]$ at $N+1$ sample points $t_0, \dots, t_N$. Using a linear multistep formula (e.g. Backward Euler) and imposing the periodicity constraint on $\mathbf{h}(s, t)$, i.e., $\mathbf{h}(s, t_0) = \mathbf{h}(s, t_N)$, we get a discretized version of the differential operator in (7.5) in terms of samples,

$$[s\mathcal{C}_{TD} + (\mathcal{G}_{TD} + \Delta\mathcal{C}_{TD})] \mathbf{H}_{TD}(s) = \mathbf{B}_{TD} \quad (7.17)$$

where

$$\mathcal{C}_{TD} = \begin{pmatrix} \mathbf{C}(t_0) & & & \\ & \mathbf{C}(t_1) & & \\ & & \ddots & \\ & & & \mathbf{C}(t_N) \end{pmatrix} \quad (7.18)$$

$$\mathcal{G}_{TD} = \begin{pmatrix} \mathbf{G}(t_0) & & & \\ & \mathbf{G}(t_1) & & \\ & & \ddots & \\ & & & \mathbf{G}(t_N) \end{pmatrix} \quad (7.19)$$

$$\Delta = \begin{pmatrix} \frac{1}{\delta_1} I & & & -\frac{1}{\delta_1} I \\ -\frac{1}{\delta_2} I & \frac{1}{\delta_2} I & & \\ & & \ddots & \\ & & -\frac{1}{\delta_N} I & \frac{1}{\delta_N} I \end{pmatrix} \quad (7.20)$$

$$\begin{aligned} \mathbf{H}_{TD}(s) &= [\mathbf{H}(s, t_0), \dots, \mathbf{H}(s, t_N)] \\ \delta_i &= t_i - t_{i-1} \\ \mathbf{B}_{TD} &= [\dots, \mathbf{b}^T, \mathbf{b}^T, \mathbf{b}^T, \dots]^T \end{aligned} \quad (7.21)$$

7.3.3 Model-Reduction Using Block-Krylov Methods

From the analysis in the previous subsections, it can be seen that both (7.12) and (7.17) can be put in the form

$$\mathbf{H}(s) = [s\mathcal{C} + \mathcal{J}]^{-1} \mathbf{B} = [I - s\mathcal{A}]^{-1} \mathcal{R} \quad (7.22)$$

where

$$\mathcal{A} = -\mathcal{J}^{-1}\mathcal{C} \quad \mathcal{R} = \mathcal{J}^{-1}\mathbf{B}. \quad (7.23)$$

Similarly $\Phi(s, t)$ can be represented by $\Phi(s)$ where

$$\Phi(s) = \mathcal{D}^T \mathbf{H} = \mathcal{D}^T [I - s\mathcal{A}]^{-1} \mathcal{R} \quad (7.24)$$

Block-Krylov subspace methods are then used to obtain reduced order approximation of $\Phi(s)$. A typical block-Krylov subspace method is run for q steps with the matrices $\mathcal{A}$, $\mathcal{D}$ and $\mathcal{R}$ as input to produce an $q \times N$ orthonormal matrix $\mathbf{V}$ where

$$\mathbf{V} = \text{colspan} [\mathcal{R}, \mathcal{A}\mathcal{R}, \mathcal{A}^2\mathcal{R}, \dots, \mathcal{A}^{(q-1)}\mathcal{R}] \quad (7.25)$$

A q -th order approximant, $\Phi_q(s)$, for $\Phi(s)$ is then obtained by

$$\Phi_q(s) = D_q^T [I_{q \times q} - sT_{q \times q}]^{-1} R_q \quad (7.26)$$

where $T_{q \times q} \in \mathbb{C}^{q \times q}$, $D_q, R_q \in \mathbb{C}^{N \times q}$, and N is the size of the original system in (7.24).

7.3.4 Difficulties in Algebraic-Based Model Reduction

It can be seen from the above discussion that existing approaches hinges on transforming the set of differential equations describing $h(s, t)$ into a system of algebraic equations to which variety of model reduction techniques based on Krylov subspace methods can be applied easily. However, this algebraic-based approach presents some practical difficulties.

For example, in the time-domain matrix formulation of the algebraic equations, one has to be careful in choosing the time step of the sampling grid. This means that portions of the waveform showing fast variations has to be over sampled as compared to other relatively slower portions. However, finding a suitable sampling grid that best approximates a yet-to-be-determined waveform is a challenging task.

Moreover, the problem of coming up with a proper sampling grid becomes even more challenging during the orthogonalization process that takes place while constructing the matrix V in (7.25). The reason for that may be explained by inspecting what happens during generating a given column of V , say v_q . During that process, the following system of equations needs to be solved first:

$$[s_0 C_{TD} + (\mathcal{G}_{TD} + \Delta C_{TD})] \tilde{v}_q = -C_{TD} v_{q-1} \quad (7.27)$$

In fact the above system corresponds to solving a steady-state problem of a linear system in which the input excitation is given by the waveform constructed from the time samples of $[-C(t_0)v_{q-1}(t_0), \dots, -C(t_N)v_{q-1}(t_N)]$ while the output response is obtained as the sample points of the vector $\tilde{v}_q$. However, this setting represents a paradoxical situation since it tries to

solve the above steady-state problem via collocation techniques and using the same sampling grid that was used at the very first iteration of the orthogonalization process. In fact, experience shows that as we go higher in the orthogonalization process, intermediate signals become faster. Fig. 7.5 shows the steady-state response for the system in (7.27) for two different signals obtained at two steps of the orthogonalization process for a Mixer test circuit. The upper graph shows the response obtained at $q = 0$ and the lower one shows the response for $q = 20$. It is clear that signals at higher-order require a more refined sampling grid than the one that was deemed sufficiently accurate at the beginning of the orthogonalization.

Figure 7.5: Steady-state response for the system (7.27), at $q = 0$ (upper graph) and $q = 20$ (lower graph)

On the other hand, the main difficulty of the frequency-domain matrix formulation is mainly computational. As seen from the analysis in Section 7.3.1, constructing the orthonormal basis V requires a matrix-vector product with $\mathcal{C}_{FD}$ and linear system solutions with $\mathcal{J}_{FD}$. These matrices are in general large and difficult to factorize. Iterative techniques [57] have been proposed to solve equations with large sparse systems efficiently. However, fast convergence of these methods requires the use of a pre-conditioner matrix that approximates the inverse of the coefficients matrix, $\mathcal{J}_{FD}$. Efficient preconditioning can be accomplished easily for diagonally-dominant matrices. However, for many systems arising from linearizing strongly nonlinear circuits around their periodic solution, diagonal dominance is not possible, since a large number of harmonics is required to linearize the system.

7.4 Proposed Algorithm

7.4.1 Elements from Hilbert Space

Let $\mathcal{U}_p(0, T)$ represent the space of continuous functions $f : \mathbb{R} \rightarrow \mathbb{C}^p$ with $f(t) = \{f_1(t), \dots, f_p(t)\}$ where each $f_k(t) \in L^2(0, T)$ (Hilbert space of all square integrable functions from 0 to T [93]). Let $\|\cdot\| : \mathcal{U}_p(0, T) \rightarrow \mathbb{C}$ described by

$$\|\mathbf{u}(t)\| = \frac{1}{T} \int_0^T \mathbf{u}(t)^{T*} \mathbf{u}(t) dt \quad (7.28)$$

define the norm on $\mathcal{U}_p(0, T)$. Denote by $\mathcal{L}_p(0, T)$ a subset of $\mathcal{U}_p(0, T)$ defined by

$$\mathcal{L}_p(0, T) = \{\mathbf{u}(t) \in \mathcal{U}_p(0, T) : \|\mathbf{u}(t)\| < \infty\} \quad (7.29)$$

and define the inner product on $\mathcal{L}_p(0, T)$ by $\langle \cdot | \cdot \rangle : \mathcal{L}_p(0, T) \times \mathcal{L}_p(0, T) \rightarrow \mathbb{C}$ where

$$\langle \mathbf{u}(t) | \mathbf{v}(t) \rangle = \frac{1}{T} \int_0^T \mathbf{u}(t)^{T*} \mathbf{v}(t) dt \quad (7.30)$$

Theorem 7.4.1 *The space $\mathcal{L}_p(0, T)$ with the inner product induced by (7.30) is a Hilbert (complete) space.*

Proof To prove that $\mathcal{L}_p(0, T)$ is a complete space, we need to show that every Cauchy sequence in $\mathcal{L}_p(0, T)$ converges to an element in $\mathcal{L}_p(0, T)$. Let $\{F_n(t)\}, n = 1, 2, 3, \dots$ be a Cauchy sequence in $\mathcal{L}_p(0, T)$, where $F_n(t) = \{f_{n,1}(t), \dots, f_{n,p}(t)\}$. Thus there exists an integer M such that $\|F_m(t) - F_n(t)\| < \epsilon$ for every $\epsilon > 0$ and for any $m, n > M$. Hence

$$\begin{aligned}
 \|F_m(t) - F_n(t)\| &< \epsilon \\
 \frac{1}{T} \int_0^T (F_m(t) - F_n(t))^{T*} (F_m(t) - F_n(t)) dt &< \epsilon \\
 \frac{1}{T} \int_0^T \sum_{k=1}^p |f_{m,k}(t) - f_{n,k}(t)|^2 dt &< \epsilon \\
 \sum_{k=1}^p \frac{1}{T} \int_0^T |f_{m,k}(t) - f_{n,k}(t)|^2 dt &< \epsilon \\
 \frac{1}{T} \int_0^T |f_{m,k}(t) - f_{n,k}(t)|^2 dt &< \epsilon
 \end{aligned} \tag{7.31}$$

Let $f_k(t) = \lim_{m \rightarrow \infty} f_{m,k}(t)$ and denote by $F(t) = \{f_1(t), \dots, f_p(t)\}$ the convergence limit of $\{F_n(t)\}$. Hence the goal is to prove that

$$\frac{1}{T} \int_0^T F(t)^{T*} F(t) dt < \infty. \tag{7.32}$$

Let $m \rightarrow \infty$ in (7.31). Hence

$$\frac{1}{T} \int_0^T |f_k(t) - f_{n,k}(t)|^2 dt < \epsilon \tag{7.33}$$

Since

$$\begin{aligned}
 \frac{1}{T} \int_0^T \sum_{k=1}^p |f_k(t)|^2 dt &= \frac{1}{T} \int_0^T \sum_{k=1}^p |f_k(t) - f_{n,k}(t) + f_{n,k}(t)|^2 dt \\
 &< \frac{1}{T} \int_0^T \sum_{k=1}^p |f_k(t) - f_{n,k}(t)|^2 + |f_{n,k}(t)|^2 dt
 \end{aligned} \tag{7.34}$$

Since $f_{n,k}(t) \in L^2(0, T)$, then

$$\frac{1}{T} \int_0^T |f_{n,k}(t)|^2 dt < \infty \quad (7.35)$$

Substituting from (7.35) and (7.33) into (7.34) shows that

$$\frac{1}{T} \int_0^T \sum_{k=1}^p |f_k(t)|^2 dt < \infty \quad (7.36)$$

Thus,

$$\|F(t)\| < \infty \quad (7.37)$$

and hence $F(t) \in \mathcal{L}_p(0, T)$. ■

The previous theorem is important since it shows that $\mathcal{L}_p(0, T)$ has a similar structure to the Euclidian space $\mathbb{C}^n$ in that there exists an orthonormal basis $\mathbf{Q}(t) = \{\mathbf{q}_1(t), \mathbf{q}_2(t), \dots\} \in \mathcal{L}_p(0, T)$ where

$$\langle \mathbf{q}_i(t) | \mathbf{q}_j(t) \rangle = \delta_{ij} \quad (7.38)$$

Similarly, given a set of elements $(\mathbf{u}_1(t), \dots, \mathbf{u}_n(t)) \in \mathcal{L}_p(0, T)$ and $\mathbf{v}(t) \in \mathcal{L}_p(0, T)$, there exists a set of scalars $\lambda_1, \lambda_2, \dots, \lambda_n$ such that

$$\mathbf{v}(t) = \mathbf{Q}(t)\boldsymbol{\lambda} \quad (7.39)$$

where $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_n]^T$.

In the ensuing discussion, the limit of integration T in (7.28) and (7.30) is implicitly understood to refer to the oscillation period of the matrices $\mathbf{C}(t)$ and $\mathbf{G}(t)$, i.e., $\mathbf{C}(t+T) = \mathbf{C}(t)$ and $\mathbf{G}(t+T) = \mathbf{G}(t)$.

7.4.2 Construction of LTV Reduced-Order Macromodel

Let $\mathbf{h}(s, t)$ be represented by the first q terms of its Taylor's series expansion around some point $s_0 \in \mathbb{C}$ in the Laplace domain,

$$\mathbf{h}(s, t) = \sum_{i=0}^{q-1} \mathbf{m}^{(i)}(t)(s - s_0)^i \quad (7.40)$$

where $\mathbf{m}^{(i)}(t)$ is the i -th order moment of $\mathbf{h}(s, t)$ at $s = s_0$. As observed from (7.40), the moments are time-dependent and since $\mathbf{h}(s, t)$ is periodic in t , then $\mathbf{m}^{(i)}(t)$ is also periodic with the same period. Equivalently, (7.40) can be put in the following matrix form,

$$\mathbf{h}(s, t) = \mathbf{K}(t)\mathbf{l}(s) \quad (7.41)$$

where

$$\begin{aligned} \mathbf{K}(t) &= [\mathbf{m}^{(0)}(t), \dots, \mathbf{m}^{(q-1)}(t)] \\ \mathbf{l}(s) &= [1, (s - s_0), \dots, (s - s_0)^{q-1}]^T \end{aligned}$$

The first step in constructing the reduced-order model is accomplished through a change of variables that is similar in form to (7.41). In this change of variables, $\mathbf{h}(s, t)$ is replaced by $\mathbf{Q}(t)\hat{\mathbf{h}}(s)$, where $\mathbf{Q}(t)$ is an orthonormal basis for the space that is spanned by $\mathbf{m}^{(i)}(t)$, $i = 0, \dots, q-1$ and $\hat{\mathbf{h}}(s) : \mathbb{C} \rightarrow \mathbb{C}^q$. Substituting for $\mathbf{h}(s, t)$ with $\mathbf{Q}(t)\hat{\mathbf{h}}(s)$ in (7.5), pre-multiplying by $\mathbf{Q}(t)^{T*}$, integrating both sides from 0 to T w.r.t. t and dividing by T yields the following reduced-order system

$$(\hat{\mathbf{N}} + \hat{\mathbf{G}} + s\hat{\mathbf{C}})\hat{\mathbf{h}}(s) = \hat{\mathbf{b}} \quad (7.42)$$

Similarly, substituting for $\mathbf{h}(s, t)$ with $\mathbf{Q}(t)\hat{\mathbf{h}}(s)$ in (7.6) yields,

$$\hat{\Phi}(s, t) = \hat{\mathbf{d}}(t)^{T*}\hat{\mathbf{h}}(s) \quad (7.43)$$

Here the matrices $\hat{\mathbf{N}}$, $\hat{\mathbf{G}}$ and $\hat{\mathbf{C}} \in \mathbb{C}^{q \times q}$ in (7.42) are given by

$$\begin{aligned} \hat{\mathbf{N}} &= \frac{1}{T} \int_0^T \mathbf{Q}(t)^{T*} \mathbf{C}(t) \frac{d\mathbf{Q}(t)}{dt} dt \\ \hat{\mathbf{G}} &= \frac{1}{T} \int_0^T \mathbf{Q}(t)^{T*} \left(\mathbf{G}(t) + \frac{d\mathbf{C}(t)}{dt} \right) \mathbf{Q}(t) dt \\ \hat{\mathbf{C}} &= \frac{1}{T} \int_0^T \mathbf{Q}(t)^{T*} \mathbf{C}(t) \mathbf{Q}(t) dt \end{aligned} \quad (7.44)$$

and the vectors $\hat{\mathbf{b}} \in \mathbb{C}^q$ and $\hat{\mathbf{d}}(t) \in \mathcal{L}_q(0, T)$ in (7.43) are given by

$$\begin{aligned}\hat{\mathbf{b}} &= \frac{1}{T} \int_0^T \mathbf{Q}(t)^{T*} \mathbf{b} dt \\ \hat{\mathbf{d}}(t) &= \mathbf{Q}(t)^{T*} \mathbf{d}\end{aligned}\tag{7.45}$$

It is to be noted that the transformation in (7.44) is known as Integrated Congruent Transform and was introduced in [1]. Substituting from (7.42) into (7.43) yields,

$$\begin{aligned}\hat{\Phi}(s, t) &= \hat{\mathbf{d}}(t)^{T*} (\hat{\mathbf{N}} + \hat{\mathbf{G}} + s\hat{\mathbf{C}})^{-1} \hat{\mathbf{b}} \\ &= \hat{\mathbf{d}}(t)^{T*} (\hat{\mathbf{I}} + s\hat{\mathbf{M}})^{-1} \hat{\mathbf{r}}\end{aligned}\tag{7.46}$$

where

$$\hat{\mathbf{r}} = (\hat{\mathbf{N}} + \hat{\mathbf{G}})^{-1} \hat{\mathbf{b}} \quad \hat{\mathbf{M}} = (\hat{\mathbf{N}} + \hat{\mathbf{G}})^{-1} \hat{\mathbf{C}}\tag{7.47}$$

Assume that the matrix $\hat{\mathbf{M}}$ has an eigen-decomposition given by

$$\hat{\mathbf{M}} = \mathbf{P} \Lambda \mathbf{P}^{-1}\tag{7.48}$$

then

$$\hat{\Phi}(s, t) = \hat{\mathbf{l}}(t)^{T*} (\hat{\mathbf{I}} + s\Lambda)^{-1} \hat{\mathbf{s}}\tag{7.49}$$

where

$$\hat{\mathbf{l}}(t) = \mathbf{P}^T \hat{\mathbf{d}}(t)^{T*} \quad \hat{\mathbf{s}} = \mathbf{P}^{-1} \hat{\mathbf{r}}\tag{7.50}$$

Thus if $\mathbf{Q}(t)$ is an orthonormal basis for $\mathbf{m}^{(i)}(t)$, $i = 0, 1, \dots, q-1$, which are periodic with period T , then $\mathbf{Q}(t)$ is also periodic with the same period T . Thus, (7.49) implies that any LPTV system can be represented by an LTI system followed by memoryless multiplication with periodic output matrix, $\hat{\mathbf{l}}(t)^{T*}$. This result actually agrees with the observation in [12] and complies with the basic result of Floquet's theory [32] which states that any linear periodically time-varying system can have a potentially an infinite number of time-independent poles and periodically time-varying residues.

7.4.3 Moment Matching Theorem

The key point in the equivalence between the original system described in (7.5)-(7.6) and the reduced one given by (7.42)-(7.43) depends essentially on the moments of $\mathbf{h}(s, t)$, $\mathbf{m}^{(i)}(t)$. To explain the main theoretical result of this section, we substitute in (7.5) with (7.40) and equate similar powers of s ,

$$\frac{d(\mathbf{C}(t)\mathbf{m}^{(0)}(t))}{dt} + s_0\mathbf{C}(t)\mathbf{m}^{(0)}(t) + \mathbf{G}(t)\mathbf{m}^{(0)}(t) = \mathbf{b} \quad (7.51)$$

$$\frac{d(\mathbf{C}(t)\mathbf{m}^{(i)}(t))}{dt} + s_0\mathbf{C}(t)\mathbf{m}^{(i)}(t) + \mathbf{G}(t)\mathbf{m}^{(i)}(t) = -\mathbf{C}(t)\mathbf{m}^{(i-1)}(t) \quad (i \geq 1) \quad (7.52)$$

Now expand both $\Phi(s, t)$ and $\hat{\Phi}(s, t)$ as a Taylor Series w.r.t. s around some point $s = s_0$ in the Laplace domain

$$\Phi(s, t) = \sum_{i=0}^{\infty} \Psi^{(i)}(t)(s - s_0)^i \quad (7.53)$$

$$\hat{\Phi}(s, t) = \sum_{i=0}^{\infty} \hat{\Psi}^{(i)}(t)(s - s_0)^i \quad (7.54)$$

where $\Psi^{(i)}(t)$ and $\hat{\Psi}^{(i)}(t)$ are the i -th order moments for $\Phi(s, t)$ and $\hat{\Phi}(s, t)$, respectively. The following theorem gives the sufficient condition for the two systems to be equivalent.

Theorem 7.4.2 *Let $\mathbf{Q}(t)$ be the transformation used in constructing the reduced system in (7.42) and (7.43). If $\mathbf{Q}(t)$ is an orthonormal basis for $\{\mathbf{m}^{(0)}(t), \dots, \mathbf{m}^{(q-1)}(t)\}$, then*

$$\hat{\Psi}^{(i)}(t) = \Psi^{(i)}(t) \quad i = 0, 1, \dots, q-1 \quad (7.55)$$

Proof The proof is based on the following three lemmas [1]².

Lemma 7.4.3 *If $\mathbf{Q}(t)$ is orthonormal and $\mathbf{u}(t) \in \mathbf{Q}(t)$, then*

$$\frac{1}{T}\mathbf{Q}(t) \int_0^T \mathbf{Q}^{T*}(t)\mathbf{u}(t) dt = \mathbf{u}(t) \quad (7.56)$$

²For proofs of Lemma 7.4.3- 7.4.5, see [1]

Lemma 7.4.4 *If $\mathbf{m}^{(i)}(t) \in Q(t)$ then there exists $\hat{\lambda}_i$ with*

$$\mathbf{m}^{(i)}(t) = Q(t)\hat{\lambda}_i \quad (7.57)$$

such that

$$(\hat{N} + \hat{G} + s_0\hat{C})\hat{\lambda}_0 = \hat{\mathbf{b}} \quad (7.58)$$

$$(\hat{N} + \hat{G} + s_0\hat{C})\hat{\lambda}_i = -\hat{C}\hat{\lambda}_{i-1} \quad (7.59)$$

Lemma 7.4.5 *Given the condition of Theorem 7.4.2, then*

$$\left(-\hat{D}(s_0)^{-1}\hat{C}\right)^j \hat{D}(s_0)^{-1}\hat{\mathbf{b}} = \frac{1}{T}\hat{\mathbf{d}}(t)^{T*} \int_0^T Q(t)^{T*}\mathbf{m}^{(j)}(t)dt \quad (7.60)$$

where

$$-\hat{D}(s_0) = \hat{N} + \hat{G} + s_0\hat{C}. \quad (7.61)$$

Substituting from (7.40) into (7.6) and comparing with (7.53) shows that

$$\Psi^{(j)}(t) = \mathbf{d}^T \mathbf{m}^{(j)}(t) \quad (7.62)$$

Substituting from (7.42) into (7.43) and using a Taylor's series expansion at $s = s_0$ yields

$$\begin{aligned} \hat{\Phi}(s, t) &= \hat{\mathbf{d}}(t)^{T*} \left(\hat{N} + \hat{G} + s_0\hat{C}\right)^{-1} \hat{\mathbf{b}} \\ &= \hat{\mathbf{d}}(t)^{T*} \sum_{j=0}^{\infty} \left[\left(-\hat{D}(s_0)^{-1}\hat{C}\right)^j (s - s_0)^j \right] \hat{D}(s_0)^{-1}\hat{\mathbf{b}} \end{aligned} \quad (7.63)$$

Comparing (7.63) with (7.54) shows that

$$\hat{\Psi}^{(j)}(t) = \hat{\mathbf{d}}(t)^{T*} \left(-\hat{D}(s_0)^{-1}\hat{C}\right)^j \hat{D}(s_0)^{-1}\hat{\mathbf{b}} \quad (7.64)$$

Manipulating (7.64) using lemmas 7.4.3- 7.4.5

$$\begin{aligned} \hat{\Psi}^{(j)}(t) &= \frac{1}{T}\hat{\mathbf{d}}(t)^{T*} \int_0^T Q(t)^{T*}\mathbf{m}^{(j)}(t)dt \\ &= \frac{1}{T}\mathbf{d}^T Q(t) \int_0^T Q(t)^{T*}Q(t)\hat{\lambda}_j dt \\ &= \mathbf{d}^T Q(t)\hat{\lambda}_j \\ &= \mathbf{d}^T \mathbf{m}^{(j)}(t) \end{aligned} \quad (7.65)$$

Hence, from (7.65) and (7.62), the proof can be obtained. ■

7.5 Orthonormal Basis Construction

The discussion of the previous section demonstrated that constructing the reduced-order LTV system is achievable if an orthonormal basis for the time-varying moments of $\mathbf{h}(s, t)$ is computed. Indeed, the approach to compute this orthonormal basis is a crucial key to the efficiency of the proposed algorithm.

As can be observed from (7.51) and (7.52), the moments are related recursively through a differential operator. Constructing the basis $\mathbf{Q}(t)$ can generally be done in two approaches. The first approach is based on the fact that $\mathbf{m}^{(i)}(t)$ is periodic in t , and hence steady-state approaches can be invoked in a manner similar to that described in Section 7.3. However, this approach requires dealing with large matrix equations. The second approach, which is adopted in this chapter, avoids completely working with large matrix equations. Instead it performs numerical time-domain integration on the systems in (7.51) and (7.52) until the steady-state point of $\mathbf{m}^{(i)}(t)$ is reached. However, applying this approach in that brute-force manner can suffer three disadvantages.

- **Ill-conditioning:** Explicit computation of $\mathbf{m}^{(i)}(t)$ is an ill-conditioned process which makes higher-order moments of practically no value for the construction of the orthonormal basis.
- **Long transients:** Numerical time-domain integration can be computationally expensive, especially for circuits with low damped time constants.
- **Numerical noise:** Numerical time-domain integration can introduce inaccuracies due to truncation errors [33]. This can degrade the accuracy of orthogonal basis and render the reduced-order model inaccurate.

The following subsections describe how the proposed algorithm avoids each one of these problem.

7.5.1 Basis Construction via Implicit Moment Computation

Constructing the orthogonal basis $Q(t)$ without computing the moments $m^{(i)}(t)$ explicitly is pivotal in obtaining accurate reduced-order LTV macromodels. This objective is commonly known as *implicit moment matching* [1]. For example, in LTI systems this is always possible using Krylov subspace methods since the moments are related through an *algebraic* operator and any standard orthogonalization process (e.g. Modified Gram-Schmidt) can be adapted to avoid computing the moments explicitly.

In this section, we present a new orthogonalization process aimed at computing an orthonormal basis for a set of time-varying moments which are related through a *differential* operator, without the need to have these moments in explicit form. To explain how the procedure works, we assume that k basis elements $\{q^{(0)}(t), \dots, q^{(k-1)}(t)\}$ have already been computed and the objective is to compute the next element, $q^{(k)}(t)$. Firstly we compute the steady-state response of the system in (7.52) due to the input $q^{(k-1)}(t)$,

$$\frac{d(C(t)\xi_k(t))}{dt} + s_0 C(t)\xi_k(t) + G(t)\xi_k(t) = -C(t)q^{(k-1)}(t). \quad (7.66)$$

Here finding the steady-state response $\xi_k(t)$ is done through integrating the system in (7.66) until all transient dies out. Using FFT, $\xi_k(t)$ can be written in its Fourier series representation,

$$\xi_k(t) = \sum_{h=-H}^H \Xi(h) e^{jh\omega_0 t} \quad (7.67)$$

Let $\tilde{\nu}_k \in \mathbb{C}^{(2H+1)n}$ be the vector obtained from

$$\tilde{\nu}_k = \sum_{h=1}^{2H+1} e_h \otimes \Xi_k(h - H - 1) \quad (7.68)$$

where e_h is the h -th column of the identity matrix and $\otimes$ denotes the Kronecker product operator. Next, $\tilde{\nu}$ is orthogonalized with respect to the column vectors $\nu_0, \dots, \nu_{k-1}$, where

$$\nu_j = \sum_{h=1}^{2H+1} e_h \otimes \Xi_j(h - H - 1) \quad j = 0, \dots, k-1 \quad (7.69)$$

and $\Xi_j(h)$ are the Fourier coefficients of $q^{(j)}(t)$. Let ν_k denote the resulting normalized vector of the orthogonalization process. $q^{(k)}(t)$ is then obtained from

$$q^{(k)}(t) = \sum_{h=1}^{2H+1} \Psi_h \nu_k e^{j(h-H-1)\omega_0 t} \quad (7.70)$$

where Ψ_j is a selector matrix defined by

$$\Psi_j = [0, \dots, \underbrace{I}_{j\text{-th position}}, \dots, 0] \quad (7.71)$$

7.5.2 Transient Compression

It is clear from the previous subsection that the main computational cost involved in constructing the basis $Q(t)$ is that of performing a time-domain integration on the system in (7.66) until the steady-state point is reached. This can, however, be very costly especially for circuits that have very long time constants.

To overcome this problem, we employ the idea of expanding the transfer function $h(s, t)$ in the right half-plane of the s -domain. This is done by choosing an expansion point $s_0 = \sigma_0 + j\omega_0$ with $\sigma_0 > 0$. This in effect, shifts all the Floquet exponents of the system far in the left-half plane and leaves the system with very short time constants. This is stated formally by the next two theorems. The first theorem can be found in [101].

Theorem 7.5.1 *Let $\psi(t)$ be a fundamental matrix of the periodical LTV system defined by an n -dimensional system of DAE,*

$$C(t)\dot{x}(t) + G(t)x(t) = 0 \quad (7.72)$$

where the matrix $C(t)$ is of rank $m \leq n$ and let T denote the period of the system. Then there exist two periodic nonsingular matrices $U(t)$ and $V(t)$ with period T such that

$$\psi(t) = U(t)D(t)V(0)C(0) \quad (7.73)$$

where

$$D(t) = \text{diag}[\exp(\mu_1 t), \dots, \exp(\mu_m t), \underbrace{0, \dots, 0}_k] \quad (7.74)$$

$$V(t)C(t)U(t) = \begin{pmatrix} I_m & 0 \\ 0 & 0 \end{pmatrix} \quad (7.75)$$

μ_i are called the characteristic (Floquet) exponents of the system in (7.72). It is obvious from Theorem 7.5.1 that the Floquet exponents control the decay of the transient response of (7.72).

Theorem 7.5.2 *Let $\psi(t)$ be a fundamental matrix for the system in (7.72), where $\psi(t) = U(t)D(t)V(0)C(0)$ as described in Theorem 7.5.1. Then the system,*

$$C(t)\dot{x}(t) + (G(t) + \sigma C(t))x(t) = 0 \quad (7.76)$$

has a fundamental matrix $\psi_\sigma(t)$ given by

$$\psi_\sigma(t) = U(t)D_\sigma(t)V(0)C(0) \quad (7.77)$$

where

$$D_\sigma(t) = \text{diag}[\exp((\mu_1 - \sigma)t), \dots, \exp((\mu_m - \sigma)t), \underbrace{0, \dots, 0}_k] \quad (7.78)$$

Proof Let μ be a diagonal matrix defined by

$$\mu = \text{diag}[\mu_1, \dots, \mu_m] \quad (7.79)$$

Substituting $\psi_\sigma(t)$ for $x(t)$ in the right-hand side of (7.76)

$$\begin{aligned}
& C(t)\dot{U}(t) \begin{pmatrix} \exp(\mu - \sigma I)t & 0 \\ 0 & 0 \end{pmatrix} V(0)C(0) + \\
& C(t)U(t) \begin{pmatrix} (\mu - \sigma I) \exp(\mu - \sigma I)t & 0 \\ 0 & 0 \end{pmatrix} V(0)C(0) + \\
& G(t)U(t) \begin{pmatrix} \exp(\mu - \sigma I)t & 0 \\ 0 & 0 \end{pmatrix} V(0)C(0) + \\
& \sigma C(t)U(t) \begin{pmatrix} \exp(\mu - \sigma I)t & 0 \\ 0 & 0 \end{pmatrix} V(0)C(0) = \\
& \left[C(t)\dot{U}(t) \begin{pmatrix} \exp(\mu t) & 0 \\ 0 & 0 \end{pmatrix} + \right. \\
& \left. C(t)U(t) \begin{pmatrix} \mu \exp(\mu t) & 0 \\ 0 & 0 \end{pmatrix} + G(t)U(t) \begin{pmatrix} \exp(\mu t) & 0 \\ 0 & 0 \end{pmatrix} \right] \begin{pmatrix} \exp(-\sigma I t) & 0 \\ 0 & 0 \end{pmatrix} V(0)C(0)
\end{aligned} \tag{7.80}$$

Since $\psi(t)$ is a fundamental matrix for the system (7.72), then it satisfies its system equations.

Thus substituting $\psi(t)$ in (7.72) yields,

$$\begin{aligned}
& \left[C(t)\dot{U}(t) \begin{pmatrix} \exp(\mu t) & 0 \\ 0 & 0 \end{pmatrix} + C(t)U(t) \begin{pmatrix} \mu \exp(\mu t) & 0 \\ 0 & 0 \end{pmatrix} + \right. \\
& \left. G(t)U(t) \begin{pmatrix} \exp(\mu t) & 0 \\ 0 & 0 \end{pmatrix} \right] V(0)C(0) = 0 \tag{7.81}
\end{aligned}$$

Postmultiplying (7.81) by $U(0)$ and using the result in (7.80) proves the theorem. ■

Corollary 7.5.3 *Let the set of Floquet exponents of the LPTV system $C(t)\dot{x} + G(t)x(t) = 0$ be*

given by $\lambda_1, \dots, \lambda_m$. Then the Floquet exponents of the system $C(t)\dot{x} + (G(t) + \sigma C(t))x(t) = 0$, are given by $\lambda_1 - \sigma, \dots, \lambda_m - \sigma$.

Thus it can be seen that choosing an expansion point in the right half-plane of the s -domain effectively shifts the Floquet exponents of (7.66) and eventually leads to a compressed transient response.

7.5.3 Eliminating Numerical Noise

The process of finding the steady-state response for the system in (7.66) through time-domain integration and using FFT to obtain a Fourier series representation as in (7.67) can become a source of significant numerical error [51]. This error can degrade the accuracy of the orthonormal basis $Q(t)$ and render the reduced model inaccurate. We present here a simple correction step to eliminate this source of error.

Assume that the system in (7.66) is solved using time-domain integration, and let $\tilde{\nu}$ be the vector comprised of the harmonics of the steady-state solution as in (7.68). (7.66) can be expressed in the frequency-domain as

$$(s_0 C_{FD} + (\mathcal{G}_{FD} + \Omega C_{FD})) \tilde{\nu} = \mathcal{U} + \Delta \quad (7.82)$$

where C_{FD} , $\mathcal{G}_{FD}$ and Ω are given as in (7.15)-(7.15); $\mathcal{U}$ represents the harmonics of solution vector in the right-hand side of (7.66) and Δ is an error term resulting from numerical errors present in the harmonics solution vector $\tilde{\nu}$. The goal of the correction step now is to compute a correction $\Delta \tilde{\nu}$ that can be used to modify $\tilde{\nu}$ so that the error term Δ is eliminated. This goal can be formulated as the solution of the following system,

$$(s_0 C + (\mathcal{G} + \Omega C)) \Delta \tilde{\nu} = \Delta \quad (7.83)$$

As mentioned earlier solving this system is a challenging task as the coefficient matrix can be

dense and coupled. Nevertheless, the solution can be obtained in a much easier way if (7.83) is converted to its time-domain form,

$$\frac{d(C(t)\zeta(t))}{dt} + s_0 C(t)\zeta(t) + G(t)\zeta(t) = \delta(t). \quad (7.84)$$

where the steady-state response $\zeta(t)$ is obtained through numerical time-domain integration. This process of correcting the original response of (7.66) is typically run for 3-5 times to obtain an extremely accurate result.

7.5.4 Summary

Figure 7.6 presents an algorithmic pseudo-code description for generating the basis $Q(t)$. Procedures “ORTHOGONALIZE” and “NORMALIZE” are responsible for orthogonalizing new members of the basis and procedure “CORRECTERROR” eliminates the numerical noise resulting from time-domain integration. Note that throughout the algorithm, generating $Q(t)$ is performed without computing the moments at any point. We are now in a position to state the main theoretical result of this section.

Lemma 7.5.4 *For mathematical convenience, denote by $Q_k(t)$ the $(k + 1)$ -th order basis $\{q^{(0)}(t), \dots, q^{(k)}(t)\}$. Let $u(t), v(t) \in \mathcal{L}_p(0, T)$ be periodic with period T and Fourier series coefficients $U(m)$ and $V(m)$, respectively, where $m = -H, \dots, H$, then*

$$\langle u(t) | v(t) \rangle = \left\langle \sum_{n=1}^{2H+1} e_n \otimes U(n - H - 1), \sum_{m=1}^{2H+1} e_m \otimes V(m - H - 1) \right\rangle \quad (7.85)$$

where $\langle \cdot, \cdot \rangle$ is the known inner-product mapping defined on the Euclidian space $\mathbb{C}^{2H+1}$.

Proof Substituting with Fourier series expansion of $u(t)$ and $v(t)$ into (7.30) and manipulating

Algorithm 7.5.1: ORTHOBASIS($G(t), C(t), b, d, s_0, \omega_o, q$)

```

main
   $Q(t) \leftarrow \phi$ 
   $m^{(0)}(t) \leftarrow \text{SOLVEPLTVS}(G(t), C(t), s_0, \omega_o, b)$ 
   $m^{(0)}(t) \leftarrow \text{CORRECTERROR}(G(t), C(t), \omega_o, b)$ 
   $q^{(0)}(t) \leftarrow \text{NORMALIZE}(m^{(0)}(t))$ 
  for  $k \leftarrow 1$  to  $q - 1$ 
    do
       $\mu(t) \leftarrow -C(t)q^{(k-1)}(t)$ 
       $\xi_k(t) \leftarrow \text{SOLVEPLTVS}(G(t), C(t), s_0, \omega_o, \mu(t))$ 
       $\xi_k(t) \leftarrow \text{CORRECTERROR}(G(t), C(t), s_0, \omega_o, \mu(t), \xi(t), \epsilon)$ 
       $\xi_k(t) \leftarrow \text{ORTHOGONALIZE}(Q(t), \xi_k(t), H, \omega_o)$ 
       $q^{(k)}(t) \leftarrow \text{NORMALIZE}(\xi_k(t))$ 
       $Q(t) \leftarrow Q(t) \cup q^{(k)}(t)$ 
  return ( $Q(t)$ )

procedure CORRECTERROR( $G(t), C(t), s_0, \omega_o, u(t), x(t), \epsilon$ )
   $\mathcal{U} \leftarrow \text{GETFD}(u(t), H), \quad \mathcal{X} \leftarrow \text{GETFD}(x(t), H)$ 
   $\Delta \leftarrow (s_0 C_{FD} + (\mathcal{G}_{FD} + \Omega C_{FD})) \mathcal{X} - \mathcal{U}$ 
  while  $\|\Delta\| > \epsilon$ 
    do
       $\delta(t) \leftarrow \text{GETTD}(\Delta, H, \omega_o)$ 
       $\xi(t) \leftarrow \text{SOLVEPLTVS}(G(t), C(t), s_0, \omega_o, \delta(t))$ 
       $\mathcal{X} \leftarrow (\mathcal{X} - \text{GETFD}(\xi(t), H))$ 
       $\Delta \leftarrow (s_0 C_{FD} + (\mathcal{G}_{FD} + \Omega C_{FD})) \mathcal{X} - \mathcal{U}$ 
  return ( $\xi(t)$ )

procedure ORTHOGONALIZE( $Q(t), \zeta(t), H, \omega_o$ )
   $\Theta \leftarrow \text{GETFD}((\zeta(t)), H)$ 
  for each  $q^{(i)}(t) \in Q(t)$ 
    do
       $V^{(i)} \leftarrow \text{GETFD}(q^{(i)}(t), H)$ 
       $\Theta \leftarrow \Theta - \langle \Theta^{T*}, V^{(i)} \rangle V^{(i)}$ 
  return ( $\text{GETTD}(\Theta, H, \omega_o)$ )

procedure NORMALIZE( $\zeta(t), H, \omega_o$ )
   $\Theta \leftarrow \text{GETFD}((\zeta(t)), H)$ 
   $\Theta \leftarrow \Theta / \|\Theta\|$ 
  return ( $\text{GETTD}(\Theta, H, \omega_o)$ )

procedure SOLVEPLTVS( $G(t), C(t), s_0, \omega_o, u(t)$ )
  Perform numerical time-domain integration on the system
   $\frac{d(C(t)\zeta(t))}{dt} + s_0 C(t)\zeta(t) + G(t)\zeta(t) = u(t)$ 
  until steady-state solution is obtained
  return ( $\zeta(t)$ )

procedure GETFD( $x(t), H$ )
  Using FFT  $x(t) \rightarrow \sum_{h=-H}^H \Xi(h) e^{j h \omega_o t}$ 
   $\mathbf{X} \leftarrow \sum_{h=1}^{2H+1} e_h \otimes \Xi(h - H - 1)$ 
  return ( $\mathbf{X}$ )

procedure GETTD( $\mathbf{X}, H, \omega_o$ )
  for  $j \leftarrow 1$  to  $2H + 1$ 
    do  $\Psi_j \leftarrow [0, \dots, \underbrace{I}_{j\text{-th position}}, \dots, 0]$ 
   $x(t) \leftarrow \sum_{h=1}^{2H+1} \Psi_h \mathbf{X} e^{j(h-H-1)\omega_o t}$ 
  return ( $x(t)$ )

```

Figure 7.6: Pseudo-code description for generating the orthonormal basis $Q(t)$

the resulting expression gives,

$$\begin{aligned}
\langle \mathbf{u}(t) | \mathbf{v}(t) \rangle &= \frac{1}{T} \int_0^T \mathbf{u}^{T*}(t) \mathbf{v}(t) dt \\
&= \frac{1}{T} \int_0^T \sum_{m=1}^{2H+1} \sum_{n=1}^{2H+1} U^{T*}(m-H-1) V(n-H-1) e^{j\omega_0(n-m)t} dt \\
&= \sum_{l=1}^{2H+1} U^{T*}(l-H-1) V(l-H-1) \\
&= \left(\sum_{l=1}^{2H+1} \mathbf{e}_l \otimes U(l-H-1) \right)^{T*} \left(\sum_{l=1}^{2H+1} \mathbf{e}_l \otimes V(l-H-1) \right) \\
&= \left\langle \sum_{n=1}^{2H+1} \mathbf{e}_n \otimes U(n-H-1), \sum_{m=1}^{2H+1} \mathbf{e}_m \otimes V(m-H-1) \right\rangle
\end{aligned} \tag{7.86}$$

which proves the lemma. ■

Lemma 7.5.5 Let $\mathbf{q}^{(k)}(t)$ be the element generated at the k -th step of algorithm 7.5.1 and denote by Γ the operator defined as,

$$\Gamma = \left[\mathbf{C}(t) \frac{d}{dt} + s_0 \mathbf{C}(t) + \left(\frac{d(\mathbf{C}(t))}{dt} + \mathbf{G}(t) \right) \right]^{-1} \mathbf{C}(t). \tag{7.87}$$

Then,

$$\mathbf{q}^{(k)}(t) = \beta_k \left(\Gamma \mathbf{q}^{(k-1)}(t) - \sum_{j=0}^{k-1} \langle \Gamma \mathbf{q}^{(k-1)}(t) | \mathbf{q}^{(j)}(t) \rangle \mathbf{q}^{(j)}(t) \right) \quad (k > 0) \tag{7.88}$$

$$\mathbf{q}^{(0)}(t) = \gamma \mathbf{m}^{(0)}(t) \tag{7.89}$$

where β_k and γ are some scalar values.

Proof Proof of (7.88) is given via substituting back through the procedures of the algorithm.

It can be seen that

$$\mathbf{q}^{(k)}(t) = \frac{1}{\|\Theta\|} \tilde{\xi}_k(t) \tag{7.90}$$

Let $\beta_k = 1/\|\Theta\|$, and substitute for $\tilde{\xi}_k(t)$ with the output from the procedure of "ORTHOGONALIZE",

$$\begin{aligned} \frac{1}{\beta_k} \mathbf{q}^{(k)}(t) = & \sum_{h=1}^{2H+1} \Psi_h \left[\sum_{l=1}^{2H+1} \mathbf{e}_l \otimes \Xi^{(k)}(l-H-1) \right] e^{j(h-H-1)\omega_o t} \\ & - \sum_{i=0}^{k-1} \left\{ \left\langle \sum_{l=1}^{2H+1} \mathbf{e}_l \otimes \Xi^{(k)}(l-H-1), \sum_{l=1}^{2H+1} \mathbf{e}_l \otimes \Xi^{(i)}(l-H-1) \right\rangle \right. \\ & \left. \left(\sum_{h=1}^{2H+1} \Psi_h \left[\sum_{l=1}^{2H+1} \mathbf{e}_l \otimes \Xi^{(i)}(l-H-1) \right] e^{j(h-H-1)\omega_o t} \right) \right\} \end{aligned} \quad (7.91)$$

where $\Xi^{(k)}(h)$ are the Fourier coefficients of $\xi_k(t)$ and $\Xi^{(i)}(h)$ are the Fourier coefficients of $\mathbf{q}^{(i)}(t)$ with $h = -H, \dots, H$ and $i = 0, \dots, k-1$. Noting that,

$$\begin{aligned} \sum_{h=1}^{2H+1} \Psi_h \left[\sum_{l=1}^{2H+1} \mathbf{e}_l \otimes \Xi^{(k)}(l-H-1) \right] e^{j(h-H-1)\omega_o t} &= \sum_{h=-H}^H \Xi^{(k)}(h) e^{jh\omega_o t} \\ &= \xi_k(t) \\ &= \Gamma \mathbf{q}^{(k-1)}(t) \end{aligned} \quad (7.92)$$

and using Lemma 7.5.4 reduces (7.91) to (7.88), thus proving the first part of the lemma. Proving (7.89) follows a similar argument. ■

Lemma 7.5.6 Let $\chi(t) \in \mathcal{Q}(t)$ where $\mathcal{Q}(t) = \{\mathbf{q}^{(0)}(t), \dots, \mathbf{q}^{(n)}(t)\}$ is orthonormal. Then,

$$\chi(t) - \sum_{i=0}^n \langle \chi(t) | \mathbf{q}^{(i)}(t) \rangle \mathbf{q}^{(i)}(t) = 0 \quad (7.93)$$

Proof Since $\chi(t) \in \mathcal{Q}(t)$, then there exists a set of scalars $\lambda_0, \dots, \lambda_n$ such that $\chi(t) \in \mathcal{Q}(t) = \mathcal{Q}(t)\lambda$ where $\lambda = [\lambda_0, \dots, \lambda_n]^T$. Substituting, in the right hand side and using the definition

of the inner product given in (7.30) yields the following

$$\begin{aligned}
 \chi(t) - \sum_{i=0}^n \langle \chi(t) | q^{(i)}(t) \rangle q^{(i)}(t) &= Q(t)\lambda - \sum_{i=0}^n \langle Q(t)\lambda | q^{(i)}(t) \rangle q^{(i)}(t) \\
 &= Q(t)\lambda - \sum_{i=0}^n \frac{1}{T} \left(\int_0^T \lambda^T Q^{T*}(t) q^{(i)}(t) dt \right) q^{(i)}(t) \\
 &= Q(t)\lambda - \underbrace{\sum_{i=0}^n (\lambda^T e_i) q^{(i)}(t)}_{Q(t)\lambda} = 0
 \end{aligned} \tag{7.94}$$

which proves the lemma. ■

Lemma 7.5.7 *Let $q^{(k)}(t)$ be the k -th member of $Q(t)$, then*

$$\Gamma q^{(k)}(t) \in Q_{k+1}(t) \tag{7.95}$$

Proof From (7.88), it can be seen that

$$\Gamma q^{(k)}(t) = \sum_{j=0}^k \langle \Gamma q^{(k)}(t) | q^{(j)}(t) \rangle q^{(j)}(t) + \frac{1}{\beta_k} q^{(k+1)}(t) \tag{7.96}$$

Hence, $\Gamma q^{(k)}(t)$ can be written as a linear combination of $\{q^{(0)}(t), \dots, q^{(k+1)}(t)\}$ and thus $\Gamma q^{(k)}(t) \in Q_{k+1}(t)$. ■

Theorem 7.5.8 *Let $Q(t)$ generated by algorithm 7.5.1 be of order q . Then $Q(t)$ is orthonormal and spans $\text{colspan} [m^{(0)}(t), \dots, m^{(q-1)}(t)]$.*

Proof Proof of this theorem is attempted by induction. From (7.89), it is straightforward to conclude that $m^{(0)}(t) \in Q(t)$. Now assume that the theorem is true for moments up to order $(k-1) < q$, i.e., $m^{(j)}(t) \in Q_{k-1}(t)$ for $j < k-1$. Define the vector

$$\rho_k(t) = m^{(k)}(t) - \sum_{j=0}^{k-1} \langle m^{(k)}(t) | q^{(j)}(t) \rangle q^{(j)}(t) \tag{7.97}$$

and define $\hat{q}^{(k)}(t) = \rho_k(t)/\|\rho_k(t)\|$. Thus,

$$m^{(k)}(t) \in \{Q_{k-1}(t) \cup \hat{q}^{(k)}\} \quad (7.98)$$

and (7.97) can be written as,

$$\|\rho_k\|\hat{q}^{(k)}(t) = \Gamma m^{(k-1)}(t) - \sum_{j=0}^{k-1} \langle \Gamma m^{(k-1)}(t) | q^{(j)}(t) \rangle q^{(j)}(t) \quad (7.99)$$

Since $m^{(k-1)}(t) \in Q_{k-1}(t)$, then there exist a set of scalars $\lambda_0, \dots, \lambda_{k-1}$ such that

$$m^{(k-1)}(t) = \sum_{i=0}^{k-1} q^{(i)}(t) \lambda_i \quad (7.100)$$

Substituting from (7.100) into (7.99) and choosing a proper arrangement of terms yields

$$\begin{aligned} \|\rho_k(t)\|\hat{q}^{(k)}(t) &= \lambda_{k-1} \Gamma q^{(k-1)}(t) - \sum_{i=0}^{k-1} \langle \lambda_{k-1} \Gamma q^{(k-1)}(t) | q^{(i)}(t) \rangle q^{(i)}(t) \\ &\quad + \left(\sum_{j=0}^{k-2} \lambda_j \left(\Gamma q^{(j)}(t) - \sum_{i=0}^{k-1} \langle \Gamma q^{(j)}(t) | q^{(i)}(t) \rangle q^{(i)}(t) \right) \right) \end{aligned} \quad (7.101)$$

By Lemma 7.5.7 $\Gamma q^{(j)}(t) \in Q_{j+1}(t)$, and by Lemma 7.5.6 the bracketed term in (7.101) vanishes. Hence, using (7.88) yields

$$\begin{aligned} \hat{q}^{(k)}(t) &= \frac{\lambda_{k-1}}{\|\rho_k(t)\|} \left(\Gamma q^{(k-1)}(t) - \sum_{i=0}^{k-1} \langle \Gamma q^{(k-1)}(t) | q^{(i)}(t) \rangle q^{(i)}(t) \right) \\ &= \eta q^{(k)}(t) \end{aligned} \quad (7.102)$$

where $\eta = \frac{\lambda_{k-1}}{\|\rho_k(t)\|\beta_k}$. Substituting (7.102) in (7.98) shows that $m^{(k)}(t) \in \{Q_{k-1}(t) \cup q^{(k)}(t)\}$, i.e., $m^{(k)}(t) \in Q_k(t)$ and thus proves the theorem. ■

7.6 Numerical Results

7.6.1 Diode Circuit

A rectifier circuit consisting of a diode and capacitors was used to test the validity of the proposed algorithm (Figure 7.7). First the steady-state response due to the input current source

shown in Figure 7.7 was computed using HB technique. The circuit was then linearized around the steady-state solution, and the resulting LPTV was considered for implementing the proposed algorithm. Figure 7.8 shows a comparison for the frequency response of the base-band frequency response of $H_1(s)$ as obtained through brute-force calculation of (7.22) and through the proposed algorithm.

Figure 7.7: A rectifier circuit.

7.6.2 Mixer Circuit

In this example, the proposed algorithm was applied to a multi-tone circuit. Consider the mixer circuit shown in figure 7.9. Steady-state solution was first obtained using HB by setting $e_1(t) = 0.15 \sin(50 \times 10^6 t)$ and $e_2(t) = 0$. Figure 7.10 shows the steady-state waveform at one of the output nodes. The circuit was then linearized around the steady-state solution and an LPTV system w.r.t. $e_2(t)$ was then constructed. The proposed algorithm was then applied to reducing the transfer function from the source $e_2(t)$ to the output node.

In order to avoid long transient response resulting from the coupling capacitors, the LPTV transfer function was expanded on the positive real axis. Figure 7.11 shows the effect of ex-

panding on the real axis. It is clear that steady-state point could be reached in only 3 cycles using expansion in the right-half plane, while expanding at the origin took several thousands of cycles. The proposed algorithm was then applied using an expansion point on the positive real axis at $s_0 = 100e^6$. Figure 7.12 shows a comparison for the frequency-domain response of $H_1(s)$ as obtained through direct computation and through an LPTV macromodel of order 20 constructed by the proposed algorithm. Here direct computation means solving the HB equations in (7.22) for all possible values of the frequency s . Figure 7.13 displays the relative percentage error in $|H_1(s)|$ obtained from the reduced system.

7.6.3 Mixer Circuit with strong nonlinearity

To demonstrate the efficiency of the proposed algorithm, the mixer circuit of the previous example was used to generate different LPTV systems with each system constructed from linearizing around a certain level of input voltage for $e_1(t)$. Table 7.1 shows a computational comparison between the proposed algorithm and the frequency-domain algebraic-based model reduction algorithm [12]. The comparison in the table is for three LPTV systems corresponding to linearizations around the steady-state solution obtained by setting the amplitudes of $e_1(t)$ to $e_1(t) = 0.15, 0.5, 1$ Volts. The table is divided into sections with each section summarizing the results obtained for the corresponding level used for $e_1(t)$. In each section, the first row displays the number of harmonics used to linearize the circuit, while the first column shows the matrix size that each algorithm worked with.

7.7 Conclusion

This chapter presented a new approach for model-reduction of LPTV system. The proposed algorithm takes advantage of the fact that for LTV systems, the transfer function has a short

Table 7.1: CPU Computational Comparison

no. of Harmonics	32			64			128		
	size	CPU	Speed Up	size	CPU	Speed Up	size	CPU	Speed Up
Algorithm from [12]	4355	554	-	8643	3360	-	17353	25200	-
Proposed Algorithm	67	49	11.3	67	183	18.4	67	541	46.7

transient response in the right-half plane of the s -domain, and hence can be obtained efficiently through time-domain integration of the system DAE equations. To avoid using explicit moment matching techniques, a variant of Krylov subspace methods have been extended to approximating transfer functions whose moments are related through a differential operator. Finally, an approach has been presented to eliminate numerical truncation errors resulting from the time-domain integration process.

(a)

(b)

Figure 7.8: Frequency response of diode circuit

Figure 7.9: A mixer Circuit

Figure 7.10: Steady-state response for the mixer circuit.

(a) Expanding at $s_0 = 0$. (b) Expanding at $s_0 = 100e^6$.

Figure 7.11: Transient response for the system in (7.51).

(a)

(b)

Figure 7.12: Frequency response of mixer circuit.

(a)

(b)

Figure 7.13: Relative percentage error in $|H_1(s)|$ obtained from the reduced system.

Chapter 8

Simulation of Nonuniform TL's via ICT

Chapter 2 highlighted the main problems of the analysis of transmission line networks. It was demonstrated that the distributed nature of these networks is naturally described in the frequency-domain using the Telegrapher's partial differential equations. However, difficulties arise in the presence of nonlinear terminations as these equations do not have a direct time-domain representation [38]. In addition, guaranteeing passivity of the models has been a major issue [15] to avoid superficial numerical oscillations.

Chapters 4 and 5 presented different approaches for the simulation of uniform distributed networks based on passive macromodelling techniques. Analysis of nonuniform TL's presents an even more challenging task. One way to obtain time-domain macromodel is to divide the line into sections and treat each section as uniform TL. Obtaining passive reduced-order time-domain macromodel can then be achieved via the various approaches explained in Chapters 4 and 5. However, this is likely to produce inaccuracies as it introduces spurious reflections at the boundaries of each section.

Another approach that is used to obtain passive reduced-order macromodel for nonuniform TL is based on the Integrated Congruence Transform (ICT) [1]. The main idea in this approach

is to construct an orthogonal basis that spans the space of the first few derivatives (moments) w.r.t. to the frequency variable of the terminal voltages and currents, and use this basis to cast Telegrapher's partial differential equations into a set of ordinary differential equations (ODE) that can be linked easily to time-domain simulators. However, constructing the orthogonal basis requires explicit computation of the moments. This explicit computation of the moments presents a numerical disadvantage since it makes higher order moments obtained from a single point expansion of practically no value in enhancing the accuracy of the reduced model to match wider frequency range. By contrast, Krylov projection model order-reduction (MOR) algorithms of lumped RLC components [81] draw their numerical robustness from the ability to construct a spanning orthogonal basis for the moments without computing the moments explicitly.

This chapter presents a new algorithm based on ICT that does not compute the moments explicitly in order to construct their spanning orthogonal basis. The proposed algorithm is developed through defining a weighted inner-product mapping on the space of the moments and then expanding the moments in polynomials that are orthonormal w.r.t. to the same weighting function. Although the proposed algorithm can handle both uniform and nonuniform TL's, the main focus of this chapter is on its application for the nonuniform case.

8.1 Outline of the Proposed Algorithm

Denote by $V(z, s)$ and $I(z, s) \in \mathbb{C}^m$ the Laplace-domain voltages and currents at an arbitrary point z along an m -conductor nonuniform transmission line of length d .

$$\begin{aligned} \frac{dV(z, s)}{dz} &= -(\mathbf{R}(z) + s\mathbf{L}(z)) \mathbf{I}(z, s) \\ \frac{dI(z, s)}{dz} &= -(\mathbf{G}(z) + s\mathbf{C}(z)) \mathbf{V}(z, s) \end{aligned} \quad (8.1)$$

where $\mathbf{R}(z)$, $\mathbf{L}(z)$, $\mathbf{G}(z)$, and $\mathbf{C}(z)$ are the per unit length (p.u.l.) space-dependent resistance, inductance, conductance and capacitance per-unit-length matrices, respectively. (8.1) can be rewritten in the following form

$$\mathbf{T} \frac{d\mathbf{X}(z, s)}{dz} = -(\mathbf{N}(z) + s\mathbf{M}(z)) \mathbf{X}(z, s) \quad (8.2)$$

where

$$\begin{aligned} \mathbf{N}(z) &= \begin{pmatrix} \mathbf{R}(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{G}(z) \end{pmatrix} \\ \mathbf{M}(z) &= \begin{pmatrix} \mathbf{L}(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(z) \end{pmatrix} \\ \mathbf{T} &= \begin{pmatrix} \mathbf{0} & \mathbf{U}_{m \times m} \\ \mathbf{U}_{m \times m} & \mathbf{0} \end{pmatrix}, \\ \mathbf{X}(z, s) &= \begin{pmatrix} \mathbf{I}(z, s) \\ \mathbf{V}(z, s) \end{pmatrix} \end{aligned} \quad (8.3)$$

and $\mathbf{U}_{m \times m}$ is an $m \times m$ identity matrix. $\mathbf{X}(z, s)$ may be expressed in a Taylor's series expansion around some frequency point, s_0 , in the frequency-domain

$$\mathbf{X}(z, s) = \sum_{i=0}^{\infty} \mathbf{W}^{(i)}(z, s_0) (s - s_0)^i \quad (8.4)$$

Substituting from (8.2) into (8.4) and equating similar powers of s yields

$$\begin{aligned} \mathcal{D}\mathbf{W}^{(0)}(z) &= \mathbf{0} \\ \mathcal{D}\mathbf{W}^{(i)}(z) &= -\mathbf{M}(z)\mathbf{W}^{(i-1)}(z) \end{aligned} \quad (8.5)$$

where $\mathcal{D}$ is an operator given by

$$\mathcal{D} \equiv \frac{d}{dz} + s_0\mathbf{M}(z) + \mathbf{N}(z) \quad (8.6)$$

It is to be noted here that the moments are related through a differential operator, while the moments in the case of lumped RLC networks are related through an algebraic operator. The objective of the proposed algorithm is to develop an orthogonalization process that generates an orthonormal basis for the set of moments, $\mathbf{W}^{(i)}(z, s_0)$, ($0 \leq z \leq d$), but without having to compute these moments explicitly.

To this end, we denote by $\mathcal{L}^{2m}(0, d)$ the space of the moments, $\mathbf{W}^{(i)}(z, s_0)$ and adopt the following version of the inner-product and norm,

$$\|\mathbf{u}(z)\| = \frac{b-a}{d} \int_a^b \mathbf{u}(\bar{z})^T \mathbf{u}(\bar{z}) w(\bar{z}) d\bar{z} \quad (8.7)$$

$$\langle \mathbf{u}(z) | \mathbf{v}(z) \rangle = \frac{b-a}{d} \int_a^b \mathbf{u}(\bar{z})^T \mathbf{v}(\bar{z}) w(\bar{z}) d\bar{z} \quad (8.8)$$

where $\bar{z} = \frac{b-a}{d}z + a$ and $w(\bar{z})$ is a positive weighting scalar function and a and b are arbitrary integration limits. It is obvious from (8.8) and (8.7) that if $\mathbf{u}(z)$ and $\mathbf{v}(z)$ are expressed as a series summation of some polynomials, $g_i(z)$,

$$\mathbf{u}(z) = \sum_{i=0}^N U_i g_i(\bar{z}) \quad \mathbf{v}(z) = \sum_{i=0}^N V_i g_i(\bar{z}) \quad (8.9)$$

where these polynomials $g_i(z)$, $i = 0, \dots, N$ are orthogonal in the interval $[a, b]$ w.r.t. to the weighting function $w(\bar{z})$, then both the inner-product norm mappings reduce to a summation of the Euclidian inner-products of the individual coefficients. The particular choice of $w(\bar{z})$, a , b and $g_i(\bar{z})$ comes as an implementation issue and will be discussed later in the next section.

Next constructing the orthogonal basis without computing the moments explicitly will be demonstrated. Denote the first k elements of the orthogonal basis by

$$\{\mathbf{q}^{(0)}(z), \dots, \mathbf{q}^{(k-1)}(z)\}.$$

Computing these elements will be explained inductively through two steps.

- **Step 1. Computing $\mathbf{q}^{(0)}(z)$.**

First the following system of differential equations is solved

$$\left(T \frac{d}{dz} + s_0 \mathbf{M}(z) + \mathbf{N}(z) \right) \tilde{\mathbf{q}}_0(z) = \mathbf{0} \quad (8.10)$$

for $\tilde{\mathbf{q}}^{(0)}(z)$. $\mathbf{q}^{(0)}(z)$ is found by normalizing $\tilde{\mathbf{q}}^{(0)}(z)$ through

$$\mathbf{q}^{(0)}(z) = \frac{1}{\|\tilde{\mathbf{q}}^{(0)}(z)\|} \tilde{\mathbf{q}}^{(0)}(z) \quad (8.11)$$

where the norm in (8.11) is obtained using (8.7).

- **Step 2. Computing $\mathbf{q}^{(i)}(z)$.**

In this step we assume that the first i , ($i < k$) elements, $\{\mathbf{q}^{(0)}(z), \dots, \mathbf{q}^{(i-1)}(z)\}$, have already been computed and the goal is to compute $\mathbf{q}^{(i)}(z)$. To this end, the following system of differential equations

$$\left(T \frac{d}{dz} + s_0 \mathbf{M}(z) + \mathbf{N}(z) \right) \tilde{\mathbf{q}}^{(i)}(z) = -\mathbf{M}(z) \mathbf{q}^{(i-1)}(z) \quad (8.12)$$

is solved for $\tilde{\mathbf{q}}^{(i)}(z, s_0)$, which is then orthogonalized w.r.t. the previous elements to produce $\hat{\mathbf{q}}^{(i)}(z)$,

$$\hat{\mathbf{q}}^{(i)}(z, s_0) = \tilde{\mathbf{q}}^{(i)}(z) - \sum_{j=0}^{i-1} \left\langle \tilde{\mathbf{q}}^{(i)}(z) | \mathbf{q}_j(z) \right\rangle \mathbf{q}^{(j)}(z) \quad (8.13)$$

where the inner product in (8.13) is obtained using (8.8). $\mathbf{q}_i(z, s_0)$ is then obtained by normalizing $\hat{\mathbf{q}}_i(z, s_0)$

$$\mathbf{q}_i(z, s_0) = \frac{1}{\|\hat{\mathbf{q}}^{(0)}(z, s_0)\|} \hat{\mathbf{q}}^{(0)}(z, s_0) \quad (8.14)$$

The basis elements computed based on the above algorithm thus spans the Hilbert subspace spanned by the first k moments $\mathbf{W}^{(i)}(z, s_0)$, i.e., there exists a set of scalars λ_j ($0 \leq j \leq k$) such that

$$\mathbf{W}^{(i)}(z, s_0) = \sum_{j=0}^k \lambda_j \mathbf{q}^{(j)}(z, s_0) \quad (8.15)$$

This result is proved mathematically in Section 8.3.

8.2 Implementation Details

This section presents further implementation details regarding the computation of the orthogonal basis $Q(z)$. In particular, Chebyshev polynomials of the first kind are adopted as our choice for the orthogonal polynomials $g_i(z)$. Under this choice, we have $a = -1$, $b = 1$ and $w(z) = 1/\sqrt{1-z^2}$. Let $u(z)$ and $v(z) \in \mathcal{L}^{2m}(0, d)$ be represented as a series of Chebyshev polynomials,

$$u(z) = \sum_{h=0}^H U_h T_h(\bar{z}) \quad v(z) = \sum_{h=0}^H V_h T_h(\bar{z}) \quad (8.16)$$

where $T_h(z)$ is the h^{th} Chebyshev polynomial of the first kind. In that case, the inner-product and norm in (8.7) and (8.8) reduce to

$$\langle u(z), v(z) \rangle = \pi V_0^T U_0 + \frac{\pi}{2} \sum_{h=1}^H U_h^T V_h \quad (8.17)$$

$$\|u(z)\| = \pi U_0^T U_0 + \frac{\pi}{2} \sum_{h=1}^H U_h^T U_h \quad (8.18)$$

Computing the k^{th} element of the basis $Q(z)$ proceeds by first representing all the z -dependent quantities in (8.12) as a summation of H Chebyshev polynomials

$$\begin{aligned} N(z) &= \sum_{h=0}^H N_h T_h(\bar{z}) \\ M(z) &= \sum_{h=0}^H M_h T_h(\bar{z}) \\ q^{(k-1)}(z) &= \sum_{h=0}^H \Xi_{k-1,h} T_h(\bar{z}) \\ \tilde{q}^{(k)}(z) &= \sum_{h=0}^H \tilde{\Xi}_{k,h} T_h(\bar{z}) \end{aligned} \quad (8.19)$$

Substituting from (8.19) into (8.12), taking the integral of both sides from -1 to $\bar{z}$, and using the following relations

$$\begin{aligned} T_m(z)T_n(z) &= \frac{1}{2}(T_{m+n}(z) + T_{|m-n|}(z)) \\ \int_{-1}^{\bar{z}} T_0(z)dz &= T_0(\bar{z}) + T_1(\bar{z}) \\ \int_{-1}^{\bar{z}} T_1(z)dz &= \frac{1}{4}(T_2(\bar{z}) - T_0(\bar{z})) \\ \int_{-1}^{\bar{z}} T_h(z)dz &= \frac{1}{2} \left(\frac{T_{h+1}(\bar{z})}{h+1} + \frac{T_{h-1}(\bar{z})}{h-1} \right) + \frac{(-1)^{h+1}}{h^2-1} \end{aligned}$$

and equating coefficients of similar Chebyshev polynomials on both sides yields

$$\left[\mathbf{A} + \frac{2}{L} (\mathbf{I}_{H+1} \otimes \mathbf{T}) \right] \tilde{\Phi}_i = \mathbf{A}_M \Phi_{i-1} \quad (8.20)$$

where

$$\begin{aligned} \mathbf{A} &= \frac{1}{2}(\mathbf{T}_1 + \mathbf{T}_2 + \mathbf{T}_3) \\ \mathbf{A}_M &= \frac{1}{2}(\mathbf{T}_{M1} + \mathbf{T}_{M2} + \mathbf{T}_{M3}) \\ \Phi_{i-1} &= [\Xi_{i-1,0}^T \cdots \Xi_{i-1,H}^T]^T \\ \tilde{\Phi}_i &= [\tilde{\Xi}_{i,0}^T \cdots \tilde{\Xi}_{i,H}^T]^T \end{aligned} \quad (8.21)$$

with

$$\mathbf{T}_1 = \begin{bmatrix} 0 & \cdots & 0 & \frac{1}{2H}D_H & \frac{1}{2H}D_{H-1} \\ 0 & \cdots & \frac{1}{2(H-1)}D_H & \frac{1}{2(H-1)}D_{H-1} & \frac{1}{2(H-1)}(D_{H-2} - D_H) \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \ddots & D_2 - \frac{1}{2}D_4 & D_1 - \frac{1}{2}D_3 & D_0 - \frac{1}{2}D_2 \\ D_H & \cdots & \cdots & \begin{pmatrix} D_1 - \frac{1}{4}D_2 + \\ \sum_{h=2}^{H-1} D_h \frac{(-1)^{h+1}}{h^2-1} \end{pmatrix} & \begin{pmatrix} D_0 - \frac{1}{4}D_1 + \\ \sum_{h=2}^H D_h \frac{(-1)^{h+1}}{h^2-1} \end{pmatrix} \end{bmatrix} \quad (8.22)$$

$$T_2 = \begin{bmatrix} \frac{1}{2H}D_1 & 0 & \cdots & \cdots & 0 & 0 \\ \frac{1}{2(H-1)}(D_2 - D_0) & \frac{1}{2(H-1)}D_1 & 0 & \cdots & 0 & 0 \\ \vdots & \cdots & \cdots & \ddots & \vdots & \vdots \\ \frac{1}{4}(D_{H-1} - D_{H-3}) & \cdots & \frac{1}{4}D_1 & \frac{1}{4}D_0 & 0 & \\ -\frac{1}{2}D_{H-2} & \cdots & -\frac{1}{2}D_1 & -\frac{1}{2}D_0 & 0 & 0 \\ \left(\begin{array}{c} -\frac{1}{4}D_{H-1}+ \\ \sum_{h=2}^H D_{H-h} \frac{(-1)^{h+1}}{h^2-1} \end{array} \right) & \cdots & \cdots & -\frac{1}{4}D_1 - \frac{1}{3}D_0 & -\frac{1}{4}D_0 & 0 \end{bmatrix} \quad (8.23)$$

$$T_3 = \begin{bmatrix} 0 & \cdots & \cdots & \frac{1}{2H}(D_{H-2} - D_H) & \frac{1}{2H}D_{H-1} \\ 0 & \ddots & \ddots & \cdots & \vdots \\ \vdots & -\frac{1}{4}D_0 & -\frac{1}{4}D_1 & \frac{1}{4}(D_0 - D_2) & \frac{1}{4}(D_1 - D_3) \\ 0 & 0 & -\frac{1}{2}D_0 & -\frac{1}{2}D_1 & D_0 - \frac{1}{2}D_2 \\ D_0 & \cdots & \sum_{h=2}^H D_{h-2} \frac{(-1)^{h+1}}{h^2-1} & \left(\begin{array}{c} -\frac{1}{4}D_0+ \\ \sum_{h=2}^H D_{h-1} \frac{(-1)^{h+1}}{h^2-1} \end{array} \right) & \left(\begin{array}{c} D_0 - \frac{1}{4}D_1+ \\ \sum_{h=2}^H D_h \frac{(-1)^{h+1}}{h^2-1} \end{array} \right) \end{bmatrix} \quad (8.24)$$

$$T_{M1} = \begin{bmatrix} 0 & \cdots & 0 & \frac{1}{2H}M_H & \frac{1}{2H}M_{H-1} \\ 0 & \cdots & \frac{1}{2(H-1)}M_H & \frac{1}{2(H-1)}M_{H-1} & \frac{1}{2(H-1)}(M_{H-2} - M_H) \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \ddots & M_2 - \frac{1}{2}M_4 & M_1 - \frac{1}{2}M_3 & M_0 - \frac{1}{2}M_2 \\ M_H & \cdots & \cdots & \left(\begin{array}{c} M_1 - \frac{1}{4}M_2+ \\ \sum_{h=2}^{H-1} M_h \frac{(-1)^{h+1}}{h^2-1} \end{array} \right) & \left(\begin{array}{c} M_0 - \frac{1}{4}M_1+ \\ \sum_{h=2}^H M_h \frac{(-1)^{h+1}}{h^2-1} \end{array} \right) \end{bmatrix} \quad (8.25)$$

$$T_{M2} = \begin{bmatrix} \frac{1}{2H}M_1 & 0 & \cdots & \cdots & 0 & 0 \\ \frac{1}{2(H-1)}(M_2 - M_0) & \frac{1}{2(H-1)}D1 & 0 & \cdots & 0 & 0 \\ \vdots & \cdots & \cdots & \ddots & \vdots & \vdots \\ \frac{1}{4}(M_{H-1} - M_{H-3}) & \cdots & \frac{1}{4}M_1 & \frac{1}{4}M_0 & 0 & \\ -\frac{1}{2}M_{H-2} & \cdots & -\frac{1}{2}M_1 & -\frac{1}{2}M_0 & 0 & 0 \\ \left(\begin{array}{c} -\frac{1}{4}M_{H-1}+ \\ \sum_{h=2}^H M_{H-h} \frac{(-1)^{h+1}}{h^2-1} \end{array} \right) & \cdots & \cdots & -\frac{1}{4}M_1 - \frac{1}{3}M_0 & -\frac{1}{4}M_0 & 0 \end{bmatrix} \quad (8.26)$$

$$T_{M3} = \begin{bmatrix} 0 & \cdots & \cdots & \frac{1}{2H}(M_{H-2} - M_H) & \frac{1}{2H}M_{H-1} \\ 0 & \ddots & \ddots & \cdots & \vdots \\ \vdots & -\frac{1}{4}M_0 & -\frac{1}{4}M_1 & \frac{1}{4}(M_0 - M_2) & \frac{1}{4}(M_1 - M_3) \\ 0 & 0 & -\frac{1}{2}M_0 & -\frac{1}{2}M_1 & M_0 - \frac{1}{2}M_2 \\ M_0 & \cdots & \sum_{h=2}^H M_{h-2} \frac{(-1)^{h+1}}{h^2-1} & \left(\begin{array}{c} -\frac{1}{4}M_0+ \\ \sum_{h=2}^H M_{h-1} \frac{(-1)^{h+1}}{h^2-1} \end{array} \right) & \left(\begin{array}{c} M_0 - \frac{1}{4}M_1+ \\ \sum_{h=2}^H M_h \frac{(-1)^{h+1}}{h^2-1} \end{array} \right) \end{bmatrix} \quad (8.27)$$

and I_{H+1} is $(H+1) \times (H+1)$ identity matrix and $\otimes$ denotes the Kronecker product operator.

The reduced system is obtained in the form

$$\hat{Y}(s) = \hat{b}^T (s\hat{M} + \hat{N})^{-1} \hat{b} \quad (8.28)$$

where

$$\begin{aligned}
\hat{M} &= \frac{d}{2} \int_{-1}^1 \mathbf{Q}(z)^T \mathbf{M}(z) \mathbf{Q}(z) dz, \\
\hat{N} &= \hat{N}_1 + \hat{N}_2, \\
\hat{N}_1 &= \frac{d}{2} \int_{-1}^1 \mathbf{Q}(z)^T \mathbf{N}(z) \mathbf{Q}(z) dz, \\
\hat{N}_2 &= \hat{T} - P, \\
P &= \mathbf{Q}_i(1)^T \mathbf{Q}_v(1) - \mathbf{Q}_i(-1)^T \mathbf{Q}_v(-1), \\
\hat{T} &= \int_{-1}^1 \mathbf{Q}(z)^T T \frac{d\mathbf{Q}(z)}{dz} dz
\end{aligned} \tag{8.29}$$

8.3 Proof of (8.15)

This section summarizes the main theoretical result of this chapter. We first start by the following two lemmas.

Lemma 8.3.1 *Let Γ denote the operator defined by*

$$\Gamma = \left[T \frac{d}{dz} + \mathbf{N}(z) + s_0 \mathbf{M}(z) \right]^{-1} \mathbf{M}(z) \tag{8.30}$$

Then,

$$\Gamma \mathbf{q}^{(k-1)}(z) \in \{\mathbf{q}^{(0)}, \dots, \mathbf{q}^{(k)}(z)\} \tag{8.31}$$

Proof Using the definition of (8.30), then (8.12) can be rewritten as

$$\tilde{\mathbf{q}}^{(k)}(z) = \Gamma \mathbf{q}^{(k-1)}(z) \tag{8.32}$$

Denote by β_k the scalar value of $1/\|\hat{\mathbf{q}}^{(k)}(z)\|$ and substitute from (8.32) into (8.13) to get

$$\frac{1}{\beta_k} \mathbf{q}^{(k)}(z) = \Gamma \mathbf{q}^{(k-1)}(z) - \sum_{j=0}^{k-1} \lambda_j \mathbf{q}^{(j)}(z) \tag{8.33}$$

where $\lambda_j = \langle \Gamma \mathbf{q}^{(k-1)}(z), \mathbf{q}^{(j)}(z) \rangle$. Hence,

$$\Gamma \mathbf{q}^{(k-1)}(z) = \frac{1}{\beta_k} \mathbf{q}^{(k)}(z) + \sum_{j=0}^{k-1} \lambda_j \mathbf{q}^{(j)}(z) \quad (8.34)$$

Thus $\Gamma \mathbf{q}^{(k-1)}(z)$ can be written as a linear combination of $\{\mathbf{q}^{(0)}(z), \dots, \mathbf{q}^{(k)}(z)\}$ which proves the lemma.

Lemma 8.3.2 *Let $\chi(z) \in \mathcal{Q}(z)$ where $\mathcal{Q}(z) = \{\mathbf{q}^{(0)}(z), \dots, \mathbf{q}^{(n)}(z)\}$ is orthonormal. Then,*

$$\chi(z) - \sum_{i=0}^n \langle \chi(z) | \mathbf{q}^{(i)}(z) \rangle \mathbf{q}^{(i)}(z) = 0 \quad (8.35)$$

Proof The proof of this lemma is similar to that of Lemma (7.94) and, hence, has been omitted.

Theorem 8.3.3 *Let $\mathcal{Q}(z)$ be the set of elements generated by executing (step 1) once and (step 2) k times as shown above. Then $\mathcal{Q}(z)$ is orthonormal and spans the Hilbert subspace spanned by the moments $\mathbf{W}^{(0)}(z), \dots, \mathbf{W}^{(k)}(z)$.*

Proof Proof of theorem 8.3.3 is given through induction. Comparing (8.5) and (8.5), it can be seen that

$$\mathbf{W}^{(0)}(z) = \tilde{\mathbf{q}}^{(0)}(z) \quad (8.36)$$

and hence

$$\mathbf{W}^{(0)}(z) = \gamma_0 \tilde{\mathbf{q}}^{(0)}(z) \quad (8.37)$$

where $\gamma_0 = \|\tilde{\mathbf{q}}^{(0)}(z)\|$. Thus $\mathbf{W}^{(0)}(z) \in \{\mathbf{q}^{(0)}(z)\}$.

Next we assume that theorem 8.3.3 holds true for moments up to order $(j-1)$, i.e.,

$$\mathbf{W}^{(j-1)}(z) \in \{\mathbf{q}^{(0)}, \dots, \mathbf{q}^{(j-1)}\} \quad j < k$$

and hence the goal now is to prove that

$$\mathbf{W}^{(j)}(z) \in \{\mathbf{q}^{(0)}, \dots, \mathbf{q}^{(j)}\}.$$

Define the element $\rho_j(z) \in \mathcal{L}(0, d)$ by

$$\rho_j(z) = W^{(j)}(z) - \sum_{i=0}^{j-1} \langle W^{(j)}(z), q^{(i)}(z) \rangle q^{(i)}(z) \quad (8.38)$$

and define $\hat{q}^{(j)}(z) = \rho_j(z) / \|\rho_j(z)\|$. Thus,

$$W^{(j)}(z) \in \{\{q^{(0)}, \dots, q^{(j-1)}(z)\} \cup \hat{q}^{(j)}(z)\} \quad (8.39)$$

and (8.38) can be written as,

$$\|\rho_j(z)\| \hat{q}^{(j)}(z) = \Gamma W^{(j-1)}(z) - \sum_{i=0}^{j-1} \langle \Gamma W^{(j-1)}(z), q^{(i)}(z) \rangle q^{(i)}(z) \quad (8.40)$$

Since $W^{(j-1)}(z) \in \{q^{(0)}(z), \dots, q^{(j-1)}(z)\}$, then there exist a set of scalars $\lambda_0, \dots, \lambda_{j-1}$ such that

$$W^{(j-1)}(z) = \sum_{p=0}^{j-1} q^{(p)}(z) \lambda_p \quad (8.41)$$

Substituting from (7.100) into (7.99) and choosing a proper arrangement of terms yields

$$\begin{aligned} \|\rho_j(z)\| \hat{q}^{(j)}(z) &= \lambda_{j-1} \Gamma q^{(j-1)}(z) - \sum_{i=0}^{j-1} \langle \lambda_{j-1} \Gamma q^{(j-1)}(z), q^{(i)}(z) \rangle q^{(i)}(z) \\ &\quad + \left[\sum_{p=0}^{j-2} \lambda_p \left(\Gamma q^{(p)}(z) - \sum_{i=0}^{j-1} \langle \Gamma q^{(p)}(z), q^{(i)}(z) \rangle q^{(i)}(z) \right) \right] \end{aligned} \quad (8.42)$$

Using Lemma 8.3.1, we get

$$\Gamma q^{(p)}(z) \in \{q^{(0)}(z), \dots, q^{(j-1)}(z)\} \quad 0 < p < j-2 \quad (8.43)$$

and hence the term is square brackets in (7.101) as a result of using Lemma 7.5.6. Thus comparing (8.33) and (7.101) yields

$$\begin{aligned} \hat{q}^{(j)}(z) &= \frac{\lambda_{j-1}}{\|\rho_j(z)\|} \left(\Gamma q^{(j-1)}(z) - \sum_{i=0}^{j-1} \langle \Gamma q^{(j-1)}(z), q^{(i)}(z) \rangle q^{(i)}(z) \right) \\ &= \eta q^{(j)}(z) \end{aligned} \quad (8.44)$$

where $\eta = \frac{\lambda_{j-1}}{\|\rho_j(z)\| \beta_j}$. Substituting (8.44) into (7.98) shows that

$$W^{(j)}(z) \in \{q^{(0)}(z), \dots, q^{(j)}(z)\}$$

and thus proves the theorem.

8.4 Numerical Results

8.4.1 Example 1

The proposed algorithm was implemented to obtain a reduced-order model for a uniform 3-conductor TL network of length $d = 10\text{cm}$. Figure 8.1 shows a graphical comparison between the exact response (obtained through an Ordinary Differential Equation (ODE) solver for the Telegrapher's equations in (8.1)) and the response obtained from the proposed algorithm after running it for $k = 65$ iterations. Also shown on the same graph, the response obtained by using 65 moments explicitly to construct the basis as in [1]. It is clear that for the same size of the reduced system, the one based on the proposed algorithm could match up double the frequency range matched through using the moments explicitly.

8.4.2 Example 2

The proposed algorithm was used to simulate a nonuniform TL network consisting of 3 coupled conductors. First the exact frequency-domain response of the system was obtained by solving the system of ordinary differential equations (ODE) in (8.1). Figures 8.2(a) and 8.2(b) show a comparison between the exact response (through ODE solution) and the response obtained from the proposed algorithm for a sample of the Y -parameters. Figure 8.3 shows a comparison for the time-domain response, due to an input step signal of 0.1 ns. The comparison is between a lumped RLC approximation of the line and the proposed algorithm. The lumped RLC model is obtained by dividing the nonuniform line into 50 sections and considering each section as a uniform line. Each uniform section is then represented by a number of lumped RLC sections according to its p.u.l. delay.

Figure 8.1: A comparison for the $|Y_{11}|$ parameter between the proposed algorithm the algorithm in [1] using explicit moments computation.

(a) Frequency Response at $Y_{1,1}$ (b) Frequency Response at $Y_{5,6}$

Figure 8.2: Numerical results for a 10 cm nonuniform transmission line. (Frequency-domain Response)

Figure 8.3: Numerical results for a 10 cm nonuniform transmission line. (Time-domain Response)

8.4.3 Example 3

A similar example to the previous one but with length, $L = 50\text{cm}$ has been employed to test the proposed algorithm. Figure 8.4 displays a frequency-domain comparison for a sample of the Y -parameters. Here the frequency-domain response obtained from the proposed algorithm was computed using only single point expansion at 1.75 GHz. Figure 8.5 displays a comparison for the time-domain response due to an input pulse of rise/fall time of 0.1 ns and a pulse-width of 8 ns.

Table 8.1 displays a CPU comparison for the time taken to simulate both lumped and reduced models for both examples. The time-domain simulation was performed using the Numerical Algorithms Group (NAG) [102] sparse differential-algebraic equations solver on a Pentium III machine.

(a) Frequency Response at $Y_{1,1}$ (b) Frequency Response at $Y_{1,6}$

Figure 8.4: Numerical results for a 50 cm nonuniform transmission line

Figure 8.5: Numerical results for a 50 cm nonuniform transmission line

	Lumped	Proposed	Speedup
10cm Line	26.3 sec	0.93 sec	28
50cm Line	693 sec	5.9 sec	117

Table 8.1: CPU time comparison

Chapter 9

Summary and Future Work

9.1 Summary

Circuit simulation has become a key cornerstone in developing efficient EDA tools. Fast and accurate circuit simulation algorithms represent a pivotal vehicle to deliver novel designs and capture the ever shrinking market windows.

Over the past decade, the concept of model reduction has become the major theme in the literature of circuit simulation due to its success in addressing the problem of managing the increasing complexity of modern circuit designs. Nevertheless, model-order reduction has been limited in application to circuits consisting of large linear blocks with few nonlinear terminations.

The work done in this proposal aims at broaching new frontiers for the applications of model-order reduction. The proposed algorithms approach that objective through generalizing projection techniques based on Krylov-subspace methods to simulate both nonlinear and linear time-varying systems.

The idea of using a generalized projection method has been first applied to the problem

of finding the steady-state solution of nonlinear circuits through projecting the large set of nonlinear equations representing the HB equations into a smaller subspace constructed from the derivatives of the unknowns with respect to some parameter. Using this projection technique enabled the reduction of the large system of equations to a much smaller set that can be solved using continuation methods.

The problem of small-signal analysis is another example where model reduction techniques can bring more efficiency to circuit simulation. Describing the behavior of small-signal in periodically excited nonlinear circuits is done through LPTV systems. Model reduction of LPTV systems has been previously based on solving a huge system of linear equations. The algorithm proposed in this work uses the idea of the compressed transient system function to avoid having to deal with a large system of equations. It rather constructs the projection subspace using transient time-domain integration for a short period of time by expanding in the right-half plane of the Laplace domain. The proposed algorithm also develops an orthogonalization procedure to construct an orthonormal basis on which the system is projected. This procedure has a numerical advantage since it does not require the elements of the subspace to be available explicitly. In addition, the approach followed in reducing the LTV system represents the natural extension of the Galerkin-type projection approach in MOR of LTI systems. However, the projection here is preformed on a generalized Hilbert space spanned by the time-dependent of the response w.r.t. the frequency variable s . This represents a shift from the projection on the classical Euclidian spaces that was used in LTI systems.

9.2 Future Work

Several research directions based on the algorithms introduced in this proposal can be investigated. As examples of these directions

- The work done in addressing model-order reduction of linear time-varying systems and nonuniform transmission lines has several applications and extensions to other areas of circuit simulation. An example of the immediate applications of the theory developed in these approaches is the problem of computing large change sensitivity of nonlinear circuits in the time-domain. This problem can be efficiently solved by noting that higher-order derivatives w.r.t. circuit parameters are related through a differential operator. These derivatives are time-dependent and may be represented as elements from a Hilbert subspace. Finding an orthogonal basis that spans this subspace and using it in an integrated Congruence transform reduces the system of nonlinear differential equations representing the circuit to a system of nonlinear algebraic equations. Large change sensitivity of the original system can be computed by sweeping the desired circuit parameter and solving the reduced system of nonlinear algebraic equations and then mapping that solution through the basis transformation operator.
- Sensitivity analysis of nonuniform transmission lines is another excellent candidate problem to be tackled using the integrated congruent transform.
- In general, various physical problems represented by partial differential equations present an interesting set of problems to be investigated using the approaches presented in Chapters 7 and 8.
- Applying the method of compressed transients to expanding the HB equations at a nonzeros value for the continuation parameter α is promising in avoiding having to deal with large system of equations at any stage in solving for the steady-state solution. This is due to the fact that the derivatives of the harmonics of the circuit variables correspond to the response of LPTV system when computed at a nonzero value for α .
- Using continuation methods to sweep different circuit parameters in the space of the

reduced system represents a very attractive way to compute the sensitivity of the steady-state solution to different design parameters. The computational efficiency is a obvious advantages since one has only to track the solution of the reduced system of nonlinear equations.

Bibliography

- [1] Q. Yu, J. M. L. Wang and E. S. Kuh, "Passive multipoint moment matching model order reduction algorithm on multiport distributed interconnect networks", *IEEE Trans. Circuits Syst. I*, vol. 46, n. 1, pp. 140–160, Jan. 1999.
- [2] T. Tang and M. Nakhla, "Analysis of high-speed VLSI interconnects using asymptotic waveform evaluation", *IEEE Trans. Computer-Aided Design*, pp. 2107–2116, Mar. 1992.
- [3] Clayton R. Paul, *Analysis of Multiconductor Transmission Lines*, Wiley, New York, 1994.
- [4] R. Achar and M. Nakhla, "Simulation of high-speed interconnects", *Proceedings IEEE*, vol. 89, n. 5, pp. 693–728, May 2001.
- [5] B. Razavi, *RF Microelectronics*, Prentice-Hall, New Jersey, 1998.
- [6] A. Dounavis, E. Gad and M. Nakhla, "Passive model-reduction of multiport distributed networks including frequency dependent parameters", *IEEE Trans. Microwave Theory Tech.*, pp. 2325–2334, December 2000.

- [7] E. Gad, A. Dounavis, M. Nakhla and R. Achar, "Passive model order reduction of multiport distributed interconnects", in *Proc. Design Automation Conf.*, pp. 526–531, Jul. 2000.
- [8] A. Dounavis, P. Gunupudi, E. Gad, R. Khazaka, R. Griffith, R. Achar and M. S. Nakhla, "Addressing transient analysis of Multiport distributed interconnects", in *Proc. IEEE conf. on Electrical Performance of Electronic Packaging*, pp. 243–246, October 2000.
- [9] E. Gad, R. Khazaka, M. Nakhla and R. Griffith, "A circuit reduction technique for finding the steady-state solution of nonlinear circuits", *IEEE Trans. Microwave Theory Tech.*, vol. 48, n. 12, pp. 2389–2396, December 2000.
- [10] E. Gad, R. Khazaka, M. S. Nakhla and R. Griffith, "A circuit reduction technique for finding the steady-state solution of nonlinear circuits", in *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 83–86, June 2000.
- [11] Joel R. Phillips, "Model Reduction of Time-Varying Linear Systems Using Approximate Multipoint Krylov-subspace Projectors", in *Proc. ACM ICCAD*, 1998.
- [12] J. Roychowdhury, "Reduced-order modeling of time-varying systems", *IEEE Trans. Circuits Syst. I*, vol. 46, n. 10, pp. 1273–1288, Oct. 1999.
- [13] E. Gad and M. Nakhla, "Efficient model reduction of linear time-varying systems via compressed transient system function", in *Proc. conf. on Design Automation and Test in Europe (DATE)*, pp. 916–922, Mar. 2002.
- [14] M. Celik, A. C. Cangellaris and A. Yaghmour, "An all purpose transmission line model for interconnect simulation in SPICE", *IEEE Trans. Microwave Theory Tech.*, vol. 45, n. 10, pp. 1857–1867, October 1997.

- [15] A Odabasioglu, M. Celik and L. T. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm", *IEEE Trans. Computer-Aided Design*, vol. 17, n. 8, pp. 645–654, August 1998.
- [16] A. Dounavis, X. Li, M. Nakhla and R. Achar, "Passive closed-loop transmission line model for general purpose circuit simulators", *IEEE Trans. Microwave Theory Tech.*, vol. 47, n. 12, pp. 2450–2459, December 1999.
- [17] K. Kundert, *The Designer's Guide to SPICE and Spectre*, Kluwer Academic Publishers, Norwell, MA., 1995.
- [18] Imad M. Jaimoukha and Ebrahim M. Kasenally, "Oblique Projection Methods for Large Scale Model Reduction", *SIAM J. Matrix Anal. Appl.*, vol. 16, n. 2, pp. 602–627, April 1995.
- [19] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis", *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, April 1990.
- [20] P. Gunupudi, R. Khazaka, A. Dounavis, Nakhla M. S. and R. Achar, "Global multi-level reduction technique for nonlinear simulation of high-speed interconnect circuits", in *Proc. IEEE conf. on Electrical Performance of Electronic Packaging*, pp. 259–262, October 2001.
- [21] P. K. Gunupudi, M. S. Nakhla and R. Achar, "Efficient simulation of high-speed distributed interconnects using krylov-subspace techniques", in *Proc. IEEE conf. on Electrical Performance of Electronic Packaging*, October 1998.
- [22] G. H. Golub and C. Van Loan, *Matrix Computations*, John Hopkins Univ. Press, Baltimore, MD, 1989.

- [23] I. M. Elfadel and David D. Ling, "A Block Rational Arnoldi Algorithm for Multipoint Passive Model-Order Reduction of Multiport RLC Networks", in *Proc. ACM ICCAD*, pp. 66–71, 1997.
- [24] Krishna Gopal, Michel S. Nakhla, Kishore Singhal and Jiri Vlach, "Distortion Analysis of Transistor Networks", *IEEE Trans. Circuits Syst.*, vol. 25, n. 2, pp. 99–106, February 1978.
- [25] Michel S. Nakhla and Jiri Vlach, "A Piecewise Harmonic-Balance Technique for Determination of Periodic Response of Nonlinear Systems", *IEEE Trans. Circuits Syst.*, vol. 23, n. 2, pp. 85–91, February 1976.
- [26] K. Kundert and A. Sangiovanni-Vincentelli, "Simulation of nonlinear circuits in the frequency domain", *IEEE Trans. Computer-Aided Design*, vol. 5, n. 4, pp. 521–535, October 1986.
- [27] Eugene L. Allgower and Kurt Georg, *Numerical Continuation Methods: An Introduction*, Springer series in computational mathematics; 13, Springer-Verlag,, 1990.
- [28] Peter Feldmann, Bob Melville and David Long, "Efficient Frequency Domain Analysis of Large Nonlinear Analog Circuits", in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 461–464, 1996.
- [29] J. Roychowdhury, "Reduced-order modelling of linear time-varying systems", in *Proc. ACM ICCAD*, pp. 92–95, Nov. 1998.
- [30] J. Roychowdhury, "Reduced-order modelling of time-varying systems", in *Proc. Asia and South Pacific Design Automation Conference*, volume 1, pp. 53–56, Wanchai, Hong Kong, Jan. 1999.

- [31] L. Zadeh, "Frequency analysis of variable networks.", in *Proc. of IRE*, pp. 291–299, Mar. 1950.
- [32] Henry D'Angelo, *Linear Time-Varying Systems: Analysis and Synthesis.*, Allyn and Bacon, Boston, MA, 1970.
- [33] Jiri Vlach and Kishore Singhal, *Computer Methods for Circuit Analysis and Design*, Van Norstrand Reinolds, New York, 1983.
- [34] C. W. Ho, A. E. Ruehli and P. A. Brennan, "The modified nodal approach to network analysis", *IEEE Trans. Circuits Syst.*, vol. 22, n. 6, pp. 504–509, June 1975.
- [35] M Nakhla and R Achar, *The VLSI Handbook*, ch. Interconnect modeling and simulation, pp. 17.1–17.29, CRC Press, Boca Raton, FL, 2000.
- [36] R. Achar, R. Khazaka and M. Nakhla, *Circuits and Systems in the Information Age*, ch. 4, pp. 187–215, IEEE Press, Piscataway, 1997.
- [37] J. R. Griffith and M. S. Nakhla, "Time-domain analysis of lossy coupled transmission lines", *IEEE Trans. Microwave Theory Tech.*, vol. 38, n. 10, pp. 1480–1487, Oct. 1990.
- [38] E. Chiprout and M. Nakhla, "Analysis of interconnect networks using complex frequency hopping (CFH)", *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 186–200, February 1995.
- [39] R. Achar, M. Nakhla and E. Chiprout, "Block CFH: A model-reduction technique for distributed interconnect networks", *IEEE Trans. Microwave Theory Tech.*, vol. 42, pp. 2562–2571, December 1994.
- [40] Thomas J. Aprille, Jr. and Timothy N. Trick, "A Computer Algorithm to Determine the Steady-State Response of Nonlinear Oscillators", *IEEE Trans. Circuit Theory*, vol. 19, n. 4, pp. 354–360, July 1972.

- [41] Thomas J. Aprille, Jr. and Timothy N. Trick, "Steady-State Analysis of Nonlinear Circuits with Periodic Inputs", *Proceedings IEEE*, vol. 60, n. 1, pp. 108–114, January 1972.
- [42] S. M. Roberts and J. S. Shipman, *Two-Point Boundary Value Problems*, Elsevier, New York, 1972.
- [43] Kenneth S. Kundert, Jacob White and Alberto Sangiovanni-Vincentelli, "A Mixed Frequency-Time Approach for Distortion Analysis of Switching Filter Circuits", *IEEE J. Solid-State Circuits*, vol. 24, n. 2, pp. 443–451, April 1989.
- [44] J. Roychowdhury, "Analyzing circuits with widely separated time scales using numerical PDE methods", *IEEE Trans. Circuits Syst. I*, vol. 48, pp. 578–594, May 2001.
- [45] Dan Feng, J. Phillips, K. Nabors, K. Kundert and J. White, "Efficient computation of quasi-periodic circuit operating conditions via a mixed frequency/time approach", in *Proc. Design Automation Conf.*, pp. 635–640, Jun. 1999.
- [46] J. Roychowdhury, "Efficient Methods for Simulating Highly Nonlinear Multi-Rate Circuits", in *Proc. Design Automation Conf.*, pp. 269–274, 1997.
- [47] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM Publishers, Philadelphia, PA, 1997.
- [48] J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, Introductory mathematics for scientists and engineers, London Wiley, 1973.
- [49] James M. Ortega, *Iterative Solution of Nonlinear Equations in Several Variables*, Computer science and applied mathematics., New York: Academic Press, 1970.

- [50] K. Mayaram, D. C. Lee, S. Moinian, D. Rich and J. Roychowdhury, "Overview of Computer-Aided Analysis Tools for RFIC Simulation: Algorithms, Features, and Limitations", in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 505–512, 1997.
- [51] K. S. Kundert, "Accurate Fourier analysis for circuit simulators", in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 25–28, May 1994.
- [52] Kenneth S. Kundert, Jacob K. White and Alberto Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*, Kluwer Academic, Boston, 1990.
- [53] Stig Skelboe, "Conditions for Quadratic Convergence of Quick Periodic Steady-State Methods", *IEEE Trans. Circuits Syst.*, vol. 29, n. 4, pp. 234–239, April 1982.
- [54] Stig Skelboe, "Computation of the Periodic Steady-State Response of Nonlinear Networks by Extrapolation Methods", *IEEE Trans. Circuits Syst.*, vol. 27, n. 3, pp. 161–175, Mar. 1980.
- [55] R. J. Gilmore and F. J. Rosenbaum, "Modeling of nonlinear distortion in GaAs MES-FETs", in *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 430–431, May 1984.
- [56] Vittorio Rizzoli and Andrea Neri, "State of the Art and Present Trends in Nonlinear Microwave CAD Techniques", *IEEE Trans. Microwave Theory Tech.*, vol. 36, n. 2, pp. 343–365, February 1988.
- [57] *Iterative Methods for Sparse Linear Systems*, PWD Publishing Company, 1996.
- [58] L. Peng Li Pileggi, "A linear-centric modeling approach to harmonic balance analysis", in *Proc. conf. on Design Automation and Test in Europe (DATE)*, pp. 634–639, 2002.

- [59] K. Kundert, "Simulation Methods for RF Integrated Circuits", in *Proc. ACM ICCAD*, pp. 752–765, 1997.
- [60] Ricardo Telichevesky, Kenneth S. Kundert and Jacob White, "Efficient Steady-State Analysis Based on Matrix-Free Krylov-Subspace Methods", in *Proc. Design Automation Conf.*, pp. 480–484, 1997.
- [61] R. Telichevesky, K. Kundert, I. Elfadel and J. White, "Fast Simulation Algorithms for RF Circuits", in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 437–444, 1996.
- [62] David Long, Robert Melville, Kirk Ashby and Brian Horton, "Full-Chip Harmonic Balance", in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 379–382, 1997.
- [63] Ricardo Telicheveski, Ken kundert and Jacob White, "Efficient AC and Noise Analysis of Two-Tone RF Circuits", in *Proc. Design Automation Conf.*, pp. 292–297, 1996.
- [64] Gerard L. G. Fokkema, D.and Sleijpen and H. A. Van der Vorst, "Accelerated Inexact Newton Schemes for Large Systems of Nonlinear Equations", *SIAM Journal on Scientific Computing*, vol. 19, n. 2, pp. 657–674, 1998.
- [65] Vittorio Rizzoli, Franco Mastri, Claudio Cecchetti and Fiorella Sgallari, "Fast and Robust Inexact Newton Approach to the Harmonic-Balance Analysis of Nonlinear Microwave Circuits", *IEEE Microwave Guided Wave Lett.*, vol. 7, n. 10, pp. 359–361, October 1997.
- [66] Vittorio Rizzoli, Franco Mastri, Fiorella Sgallari and Giulia Spaletta, "Harmonic-Balance Simulation of Strongly Nonlinear Very Large-Size Microwave Circuits by Inexact Newton Methods", in *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 1357–1360, 1996.
- [67] Stephen A. Maas, *Nonlinear Microwave Circuits*, IEEE Press, New York, 1996.

- [68] Kenneth S. Kundert, Gregory B. Sorkin and Alberto Sangiovanni-Vincentelli, "Applying Harmonic Balance to Almost-Periodic Circuits", *IEEE Trans. Microwave Theory Tech.*, vol. 36, n. 2, pp. 366–378, February 1988.
- [69] Y. Yamagami, Y. Nishio, A. Ushida, M. Takahashi and K. Ogawa, "Analysis of Communication Circuits Based on Multidimensional Fourier Transformation", *IEEE Trans. Computer-Aided Design*, vol. 18, n. 8, pp. 1165–1177, August 1999.
- [70] E. Ngoya and R. Larcheveque, "Envelop transient analysis: a new method for the transient and steady state analysis of microwave communication circuits and systems", in *IEEE MTT-S Int. Microwave Symp. Dig.*, volume 3, pp. 1365–1368, Jun 1996.
- [71] J. Roychowdhury, "MPDE methods for efficient analysis of wireless systems", in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 451–454, May 1998.
- [72] A. E Ruehli, "Equivalent circuit models for three dimensional multiconductor systems", *IEEE Trans. Microwave Theory Tech.*, vol. 22, n. 3, pp. 216–221, Mar. 1974.
- [73] H. Heeb and A. E. Ruehli, "Retarded models for PC board interconnects – or how the speed of light affects your SPICE circuit simulation", in *Proc. ACM ICCAD*, pp. 70–73, November 1991.
- [74] H. Heeb and A. E. Ruheli, "Three dimensional interconnect analysis using partial element equivalent circuits", vol. 39, pp. 974–982, 1992.
- [75] L. Weinberg, *Network Analysis and Synthesis.*, New York: McGraw-Hill., 1962.
- [76] E. S. Kuh and R. A. Rohrer, *Theory of Linear Active Networks*, Holden-Day, San Francisco, 1967.

- [77] K. J. Kerns and A. T. Yang, "Preservation of passivity during RLC network network reduction via split congruence transformations", *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 582–591, July 1998.
- [78] W. C. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers", *Journal of Applied Physics*, vol. 19, n. 1, pp. 55–63, January 1948.
- [79] Theodore J. Rivlin, *The Chebyshev Polynomials*, Number Pure and applied mathematics, New York: Wiley, 1974.
- [80] A. Dounavis, R. Achar and M. Nakhla, "A general class of passive macromodels for lossy multiconductor transmission lines", *IEEE Trans. Microwave Theory Tech.*, vol. 49, n. 10, pp. 1686–1696, October 2001.
- [81] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé via Lanczos process", *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 639–649, May 1995.
- [82] M. Sylveria, M. Kamon, I. Elfadel and J. White, "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of arbitrary RLC circuits", in *Proc. ACM ICCAD*, pp. 288–294, November 1996.
- [83] D. Kuznetsov and J. E. Schutt-Aine, "Optimal transient simulation of transmission lines", *IEEE Trans. Circuits Syst.*, vol. 43, n. 2, pp. 110–121, February 1996.
- [84] T. K. Tang, M. S. Nakhla and R. Griffith, "Analysis of lossy multiconductor transmission lines using the asymptotic waveform evaluation technique", *IEEE Trans. Microwave Theory Tech.*, vol. 39, n. 12, pp. 2107–2116, Jul. 1991.
- [85] N. W. McLachlan, *Laplace Transforms and Their Applications to Differential Equations*, New York, Dover, 1962.

- [86] W. E. Arnoldi, "The principle of minimized iteration in the solution of the matrix eigenvalue problem", *Quart. Appl. Math.*, vol. 9, pp. 17–29, 1951.
- [87] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators", *Journal of Research of the National Bureau of Standards*, vol. 45, pp. 255–282, 1950.
- [88] R. W. Freund and P. Feldmann, "The SyMPVL algorithm and its applications to interconnect simulation", in *sispad*, pp. 113–116, Sep., 1997.
- [89] A. Odabasioglu, M. Celik and L.T. Pileggi, "Practical considerations for passive reduction of RLC circuits", in *Proc. ACM ICCAD*, pp. 214–219, 1999.
- [90] M. Kamon, F. Wang and J. White, "Generating nearly optimally compact models from Krylov subspace based reduced-order models", *IEEE Trans. Circuits Syst. II*, vol. 47, n. 4, pp. 239–248, April 2000.
- [91] W. E. Arnoldi, "The principle of minimized iteration in the solution of the matrix eigenvalue problem", *Quarterly Applied Mathematics*, vol. 9, pp. 17–29, 1951.
- [92] J. M. Wang, E. S. Kuh and Qinglian Yu, "The Chebyshev expansion based passive model for distributed interconnect networks", in *Proc. ACM ICCAD*, pp. 370–375, Nov. 1999.
- [93] Lokenath. Debnath, *Introduction to Hilbert Spaces with Applications*, Academic Press, Boston, MA, 1990.
- [94] A. Dounavis, R. Achar and M. Nakhla, "Efficient passive circuit models for distributed networks with frequency-dependent parameters", *IEEE Trans. Adv. Packag.*, vol. 23, pp. 382–392, August 2000.

- [95] Anestis Dounavis, "Time Domain Macromodels for High-Speed Interconnects.", Master's thesis, Carleton Univ., 2000.
- [96] David A. Harville, *Matrix Algebra From a Statistician's Perspective*, Springer, New York, 1997.
- [97] R. Griffith and M. Nakhla, "A new high order absolutely stable explicit numerical integration algorithm for the time domain simulation of nonlinear circuits", in *Proc. ACM ICCAD*, pp. 504–509, 1997.
- [98] P. Gunupudi, R. Khazaka and M. S. Nakhla, "Analysis of Transmission Line Circuits using Multi-Dimensional Model Reduction Techniques", in *Proc. IEEE conf. on Electrical Performance of Electronic Packaging*, pp. 43–46, October 2001.
- [99] Vittorio Rizzoli, Claudio Cecchetti and Franco Mastri, "A Rigorous Frequency-Domain Approach to Large-Signal Noise in Nonlinear Microwave Circuits", *IEEE Microwave Guided Wave Lett.*, vol. 8, n. 6, pp. 220–222, June 1998.
- [100] Makiko Okumura, Hiroshi Tanimoto, Tetsuro Itakura and Tsutomu Sugawara, "Numerical Noise Analysis for Nonlinear Circuits with a Periodic Large Signal Excitation Including Cyclostationary Noise Sources", *IEEE Trans. Circuits Syst. I*, vol. 40, n. 9, pp. 581–590, September 1993.
- [101] Alper Demir and Alberto Sangiovanni-Vincentelli, *Analysis and Simulation of Noise in Nonlinear Electronic Circuits and Systems*, Kluwer Academic, Boston, 1998.
- [102] Numerical Algorithms Group Limited, Oxford, UK, *NAG Fortran Library Mark 19*, 1999.